

Optimal Space Communications Techniques

Semi-Annual Report

June 27, 1974 - December 26, 1974

Goddard Space Flight Center

Greenbelt, Maryland

under

NASA Grant NSG - 5013

(NASA-CR-142071) OPTIMAL SPACE  
COMMUNICATIONS TECHNIQUES Semiannual  
Report, 27 Jun. - 26 Dec. 1974 (City Coll.  
Research Foundation) 73 p HC \$4.25 CSCL 17B

N75-15858

Unclas

G3/32 08989

COMMUNICATIONS SYSTEMS LABORATORY

DEPARTMENT OF ELECTRICAL ENGINEERING

Donald L. Schilling

Professor of Electrical Engineering

Principal Investigator



THE CITY COLLEGE  
RESEARCH FOUNDATION

THE CITY COLLEGE of  
THE CITY UNIVERSITY of NEW YORK



Optimal Space Communications Techniques

Semi-Annual Report

June 27, 1974 - December 26, 1974

Goddard Space Flight Center

Greenbelt, Maryland

under

NASA Grant NSG - 5013

COMMUNICATIONS SYSTEMS LABORATORY  
DEPARTMENT OF ELECTRICAL ENGINEERING

Donald L. Schilling

Professor of Electrical Engineering

Principal Investigator

## Table of Contents

Intoduction

I. Video Encoding Using an Adaptive Digital Delta Modulator

II. The Effect of Channel Errors on DM Encoded Video Signals

## Introduction

This semianhual report summarizes one aspect of the research sponsored by the National Aeronautics and Space Administration under NASA Grant NSG - 5013 for the period June 27, 1974 - December 16, 1974.

The research discussed in this report considers the encoding of video signals using the Song Adaptive Delta Modulator. This DM conceived by Professor D. L. Schilling and developed by Drs. C. L. Song and J. Garodnick is widely used today for voice encoding. It is considered to be the best "sounding" of all DM systems by Engineers at the NASA JSC in Houston, Texas.

Currently research is being directed toward the operation of the Song ADM system for video transmission. Video signals can be characterized as a sequence of pulses having arbitrary height and width. The ADM is well suited to track signals having fast rise times as the DM algorithm permits an exponential rise to estimate an input step. However, the DM algorithm also results in a large overshoot and an underdamped response the step. This is analogous to the response of an RLC circuit to a step. In Part I of this report we present an Overshoot Suppression algorithm which significantly reduces the "ringing" while not affecting the rise time. Formulii for the "rise time" and "settling time" is presented.

Another problem considered in this report deals with channel errors and their effect on the DM encoded bit stream. It is well known that when a bit in a DM sequence is in error that the decoded waveform will have an error which can propagate virtually forever. Part II of this report considers two algorithms which virtually eliminate the errors.

During this report several papers were presented at conferences. Part I and II of this report will be submitted for Publication. A complete list of publication will be given in the final report.

Participating in this program are:

Drs. J. Garodnick, I. Paz and D. L. Schilling

and

Messrs. R. Lei, J. LoCicero, V. Rao, N. Scheinberg, D. Ucci

and L. Weiss

## I. Video Encoding Using an Adaptive Digital Delta Modulator

### ABSTRACT

An overshoot suppression scheme to improve the performance of the Digital-Song-Adaptive Delta Modulator (DM) for picture transmission is described. This scheme allows for a faster increase in the step sizes than permitted by overshoot and settling time considerations, thus improving rise-time performance. The overshoot suppression (OSS) algorithm used has been verified using computer simulations on a PDP-8. Furthermore, experimental results using computer generated test pictures as well as pictures from a flying-spot scanner show the improvements due to the overshoot suppression scheme. It is also shown that the additional hardware required for the actual implementation of the algorithm is simpler than those encountered in the literature, and gives better signal tracking accuracy. Upper and lower bounds for the settling time, with and without OSS are derived showing an improvement with the suppression scheme. Stability conditions are also derived permitting the proper selection of DM parameters. Considering the DM link as a nonlinear digital filter, a formula that relates the minimum rise time that can be handled for given filter parameters and voltage swings is developed. In developing the formula the problem of random truncation errors due to the finite arithmetic implementation is handled.

## I. INTRODUCTION

Figure 1 shows the structure of the digitally implemented Adaptive Delta Modulation (ADM) System referred to in this paper. Briefly, its operation is as follows:

The input signal  $S(t)$  is sampled and A to D converted to give  $S_k$ .  $S_k$  is then compared to its estimate,  $X_k$ , generating a sign-bit  $e_k$ , with

$$e_k = \text{sgn}(S_k - X_k) \quad (1)$$

where

$$X_k = X_{k-1} + \Delta_k \quad (2)$$

The step size at the  $k^{\text{th}}$  sampling instant is

$$\Delta_k = g_1(e_{k-1}, \Delta_{k-1}) + g_2(e_{k-2}, \Delta_{k-1}) \quad (3)$$

Thus the  $k^{\text{th}}$  step size depends on the previous step size, and the previous two sign bits. The  $g_1$  and  $g_2$  function characteristics, shown in Fig (2), are a generalization of the step-size algorithm of the Song ADM [1] permitting a variation of its parameters for possible optimization.

Figure (2) indicates that:

$$\Delta_k = \begin{cases} |\Delta_{k-1}| (\alpha e_{k-1} + \beta e_{k-2}) & , \quad |\Delta_{k-1}| \geq M \\ M e_{k-1} & , \quad |\Delta_{k-1}| < M \end{cases} \quad (4)$$

where the special zero region,  $|\Delta_{k-1}| < M$ , is needed to prevent a dead zone at the origin. (Note that we are assuming a digital implementation where the fractional part of  $\Delta_k$  is truncated, but the positive

constants  $\alpha$  and  $\beta$  may be nonintegral.) Based upon these considerations we define  $I = 1/(\alpha + \beta - 1)$ . Then the zero region is

$$M = \begin{cases} I & , I \text{ an integer} \\ \text{Truncation } (I + 1) & , I \text{ not an integer} \end{cases} \quad (5)$$

where the function Truncation (·) is defined to give the integer portion of its argument. A consequence of the above is that the minimum step size possible is given by

$$\Delta = \text{Truncation } ( (\alpha - \beta) M ) \quad (6)$$

Defining two new symbols,  $\gamma = \alpha + \beta$ , and  $\delta = \alpha - \beta$ , note the following specific cases. If  $\gamma \geq 2$ , then  $\Delta = 0$  and  $M = 1$ . If  $1.5 \leq \gamma < 2$ , then  $M = 2$  and  $\Delta = 1$  if  $\delta \geq 0.5$  (also  $\Delta = 0$  if  $\delta < 0.5$ ). It is apparent from this, and in general from Eq (5) and (6), that  $M$  and  $\Delta$  decrease with increasing  $\gamma$ . We show later that this interaction between  $M$ ,  $\Delta$  and  $\gamma$  is desirable in many applications such as video. Finally, note that  $\gamma$  must be greater than unity to permit an increasing step-size adaptation when operating in "slope-overload", i.e., when tracking a rapidly varying input. To permit the large step-sizes that evolve during slope-overload to decay when tracking a slowly varying input we must have  $0 < \delta < 1$ .

The decoder is the feedback portion of the encoder. It reconstructs the approximation  $X_k$  from the  $e_k$  bit stream.  $X_k$  is then D/A converted and low pass filtered to give  $\hat{s}(t)$ , the estimate of the transmitted signal.

In video processing,  $S(t)$  may contain many large discontinuities of very short rise times. This corresponds to abrupt changes in grey level in the picture content. Thus, edge response is very important in video. A linear delta modulator, however, is limited in its ability to track sudden input changes by its fixed step size, the magnitude of these steps being bounded by the permissible granular noise in the constant shade regions,



see Fig 3a. Shade contrast is thus degraded by the so called "slope-overload-noise" introduced by the DM channel. A further, and often subjectively worse degradation that is introduced is edge busyness. This is observed as a jaggedness along what should appear as smooth edges. It has been shown by Oshima and Ishiguro [2] that this effect is minimized if slope overload is not permitted. Thus, to alleviate edge busyness and permit  $\hat{S}(t)$  to approximate rapid rises, i.e., minimize the slope-overload-noise, it is desirable to make the step sizes  $\Delta_k$  small when tracking a slowly varying input but allow them to increase quickly, in some nonlinear fashion, when tracking rapidly varying inputs. This is done in the Song adaptive DM [1] by adjusting  $\alpha$ ,  $\beta$ , and  $\Delta$  to meet rise time requirements. See Fig 3b and Eqs (1) through (4). Other ADM schemes [3], [4], [5] have similar step size variability and our results and discussions are, in principle, of general applicability.

The sharp rises in a video signal are often followed by regions of fairly constant level due to regions of uniform shade in the picture. Thus, while alleviating slope overload problems, an adaptive DM introduces the possibility of large overshoots when the tracked level is finally reached. Furthermore, the overshoot is followed by a transient oscillatory response until the DM finally locks onto the tracked signal level. These effects are shown in Fig 3b. Figure 3b also shows that good steady state response, i.e., small amplitude oscillations about a constant level in  $S(t)$ , requires a small minimum step,  $\Delta$ . It will also be shown that the DM becomes unstable if  $\alpha$  and  $\beta$  are chosen improperly. Thus, in choosing  $\alpha$ ,  $\beta$ , and hence indirectly  $\Delta$ , a trade-off must be made between slope overload noise and edge busyness versus overshoots, including the recovery (settling) time and the requirements for steady state response. Moreover, all this must be done while maintaining DM stability.

We may therefore conclude as follows: Both the overshoot and the subsequent recovery time are undesirable attributes of an adaptive DM. Reducing the step sizes decreases the possible overshoot amplitudes

and shortens the recovery time. This, however, augments slope overload, edge busyness and rise time. A trade-off therefore exchanges overshoot amplitude and its recovery time for slope-overload-noise in an adaptive DM.

Overshoot suppression (OSS) is a scheme to sharply limit the overshoot amplitude and reduce the subsequent recovery time. This is done without reducing the step sizes until overshoot is imminent. Thus, slope-overload-noise, and hence the rise time, may be decreased. Using this technique  $\gamma$  can be increased while decreasing  $\Delta$ . In this way, slope overload as well as small steady state amplitude requirements can be met simultaneously and, obviously, overshoots and subsequent recovery times are minimized.

An OSS algorithm has been suggested in the literature [6] which uses a "look-up table" where arbitrary values for the step sizes are tabulated. Furthermore, the maximum step size is limited by overshoot considerations.

In the Song Delta Modulator, the step sizes can continually increase with the OSS scheme described in this paper [7] thus yielding better signal to slope-overload-noise ratio than that obtained using the other OSS technique. Moreover, the amount of equipment involved in implementing our OSS algorithm is very modest in comparison to the equipment needed to implement the other proposed scheme, and could fit into any ADM in which the next step size is explicitly calculated. It is also flexible as to the amount of OSS it can perform and trade-offs between conflicting factors can be accurately set.

The OSS algorithm proposed in this paper has been checked by computer simulations. The results obtained indicate the feasibility of increasing the rise time capability of the DM without causing excessive overshoots and subsequent oscillations. The improvements in the quality of the transmitted video is illustrated with computer generated pictures processed by the DM with and without OSS as well as by real pictures

from a flying-spot scanner.

Theoretical results include upper bounds for the settling time, with and without OSS, showing a considerable improvement with the suppression scheme. A general theoretical study is made of ADM stability as a function of the parameters  $\gamma$  and  $\delta$ . Previous investigators such as Jayant [4] only assumed a stability condition of the form  $\gamma \delta \leq 1$ . Our results are more detailed and permit the proper selection of the ADM parameters to satisfy performance trade-offs. Finally, the finite arithmetic inherent in the digital implementation of the DM results in random truncation errors in Eq (4) that cause inaccuracies in deriving equations to describe performance. However, using certain approximations a deterministic relation between the rise time and the signal level has been obtained. Computer investigations indicated that for all practical cases tested, the results obtained using this relation are in close agreement with the actual rise-times.

## II. THE PROPOSED OVERSHOOT SUPPRESSION ALGORITHM

The Overshoot Suppression Algorithm may be understood by considering the four cases shown in Fig 4 in which an overshoot or an undershoot occurs. In Fig 4a an overshoot occurs at sampling time  $k - 1$  followed immediately by an undershoot at  $k$ . For this case it is easy to show that the DM will approach its steady-state condition rapidly. This is not the case in Fig 4b where the overshoot is larger than in (a) and  $X_k$  is greater than  $S_k$ . Consequently an undershoot occurs at  $k + 1$  or later and with an amplitude larger than in (a). This occurs because the step sizes begin increasing again after the first reversed step. Thus it will take many more sampling periods to reach steady state in (b) than in (a). The algorithm is therefore implemented only when case (b) occurs. Note that Figs 4(c) and 4(d) depict undershoots corresponding to the overshoots in (a) and (b) respectively. Action to prevent excessive undershoots is thus taken only for case (d).

The occurrence of cases (b) and (d) can be recognized by examining the sequence  $\{e_{k-3}, e_{k-2}, e_{k-1}, e_k\}$ . The fingerprint of (b) is  $\{+1, +1, -1, -1\}$ , while that of (d) is  $\{-1, -1, +1, +1\}$ . When either sequence is encountered action is taken to prevent overshoot or undershoot. The corrective action entails decreasing the stored values of  $\Delta_{k-1}$ , and hence  $X_{k-1}$ , and  $X_k$ . Case (b) is thus transformed into a case (a) situation and the same for (d) and (c) respectively. The shape of the modified waveform actually depends on the amount by which  $\Delta_{k-1}$  is decreased. The simplest scheme is one where  $\Delta_{k-1}$  is replaced by half of its original value. We may allow for more rapidly increasing step sizes  $\Delta_k$ , i.e., larger  $\gamma$  (see Eq 4), as long as  $\Delta_{k-1}$  is replaced by a smaller fraction of its original value when OSS is employed.

Now the Overshoot Suppression Algorithm is applied to the Song ADM operating in the Video Mode where  $\alpha = 1$ ,  $\beta = 0.5$ . It is shown elsewhere [8] that good video transmission results using these parameters even without OSS. With the addition of the suppression algorithm video reproduction is improved.

The salient features of the Song Video Mode response are now summarized. In approaching a level from above or below as in Fig 4, each step size is 1.5 times the previous one (see Eq 4 for  $\alpha = 1$ ,  $\beta = 0.5$ ). When a direction reversal occurs, as at sampling time  $k$  in Fig 4, the first step size following the reversal is one half the previous step size, i.e.,  $\Delta_k = -1/2 \Delta_{k-1}$  (see Eq 4). Thus, in Fig 4(b) we have

$$X_k \geq X_{k-1} - 1/2 \Delta_{k-1} = X_{k-2} + 1/2 \Delta_{k-1} \quad (7)$$

The inequality sign is needed due to the fixed point arithmetic employed in the digital implementation. Also in Fig 4(b)

$X_{k-2} < S_k < X_k$ . To implement OSS, set  $(\Delta_{k-1})' = 1/2 \Delta_{k-1}$ , where the prime refers to the new values after the OSS algorithm has been implemented. Therefore

$$(X_{k-1})' = X_{k-2} + (\Delta_{k-1})' = X_{k-2} + 1/2 \Delta_{k-1} \quad (8)$$

Next set

$$(\Delta_k)' = \Delta_k = -1/2 \Delta_{k-1} \quad (9)$$

Thus

$$(X_k)' = (X_{k-1})' + (\Delta_k)' = X_{k-2} \quad (10)$$

Hence, Fig 4(b) has been transformed into Fig 5, with undershoot occurring at  $k$  rather than at  $k + 1$  or later. It should be evident from Fig 5, even without a detailed explanation of the worst case, that the overshoot has been at best entirely eliminated or at worst cut in half depending on whether  $S(t)$  is closer to  $(X_{k-1})'$  or  $X_{k-2}$  respectively. Figure 5 also shows that the recovery time is greatly reduced since the DM locks onto  $S(t)$  very rapidly after sampling time  $k$ . The improvement in recovery time due to OSS is discussed in Section VII. Note also that now  $(e_k)' = \text{sgn}(S_k - (X_k)') = 1$ , whereas in Fig 4(b)  $e_k = \text{sgn}(S_k - X_k) = -1$ .

The above OSS scheme is summarized in the form of an algorithm by considering a typical cycle of the modified DM.

Step 1: Generate  $S_k$ .

Step 2: Calculate  $\Delta_k = g_1(e_{k-1}, \Delta_{k-1}) + g_2(e_{k-2}, \Delta_{k-1})$

Step 3: Calculate  $X_k = X_{k-1} + \Delta_k$

Step 4: Calculate  $e_k = \text{sgn}(S_k - X_k)$  and transmit this bit.

In the DM without OSS this would complete the cycle. That is,  $k$  is next updated and steps 1 through 4 are repeated. To implement OSS the following additional steps are needed:

Step 5: If  $e_{k-3} = e_{k-2} = 1$ , and  $e_{k-1} = e_k = -1$ , set  $V = 1$ . If  $e_{k-3} = e_{k-2} = -1$ , and  $e_{k-1} = e_k = 1$ , set  $W = 1$ .

Step 6: If  $V \neq 1$  and  $W \neq 1$  go to Step 7. Otherwise set

- (a)  $(\Delta_{k-1})' = 1/2 \Delta_{k-1} = -\Delta_k$
- (b)  $(X_{k-1})' = X_{k-2} + (\Delta_{k-1})' = X_{k-2} + 1/2 \Delta_{k-1} = X_k$
- (c)  $(\Delta_k)' = \Delta_k = -1/2 \Delta_{k-1}$
- (d)  $(X_k)' = (X_{k-1})' + (\Delta_k)' = X_{k-2}$
- (e)  $(e_k)' = -e_k$

Step 7: Update  $k$ . That is, set  $e_{k-3} = e_{k-2}$ ;  $e_{k-2} = e_{k-1}$ . If step 6 has been executed set  $e_{k-1} = (e_k)'$ ;  $X_{k-2} = (X_{k-1})'$ , etc. Otherwise set  $e_{k-1} = e_k$ ;  $X_{k-2} = X_{k-1}$ , etc. This completes a cycle.

The OSS algorithm introduces, with a small though nonzero probability, the undesirable effect of increasing the duration of large amplitude narrow-pulses. In video, this results in a smearing effect in highly detailed picture areas. This occurs since  $S(t)$ , which does not remain constant, may start decreasing and reach a value less than  $(X_k)'$  at  $t = k$ . However, the OSS algorithm automatically sets  $(e_k)' = 1$ . Thus, it is only at  $t = k + 2$  that  $X_k$  starts following the trailing edge of the pulse. Furthermore, the initial rate of attack on the trailing edge will be very slow since  $\Delta_{k+2} = -1/8 \Delta_{k-1}$ . Whereas it is impossible to eliminate the delay in the trailing edge of the response, exceedingly slow rates of attack may be avoided by imposing a threshold on the OSS algorithm. That is, to avoid a very small  $\Delta_{k+2}$  OSS is not implemented when  $\Delta_{k-1}$  is smaller than some threshold value. Note, that this does not reduce the effectiveness of the OSS algorithm since a small  $\Delta_{k-1}$  cannot produce a large overshoot anyway. The effectiveness of the OSS algorithm combined with a threshold is shown in Section IV with ADM processed flying spot scanner pictures.

### III. HARDWARE IMPLEMENTATION OF THE OVERSHOOT SUPPRESSION ALGORITHM

The implementation of the above OSS algorithm requires the addition of very little hardware to the Digital-Song-Adaptive Delta Modulator. This can be seen by considering the schematic representation of the ADM CODEC with OSS shown in Fig 1. Note that the extra components needed to implement the suppression scheme appear in branches that are drawn with dashed lines. These elements include the delay  $D5$ , as well as the indicated gates needed for decision, switching and timing purposes.

It is difficult to discern the operation of the circuit by merely examining the schematic diagram in Fig 1 because the sequential order of operation is not specified in the diagram. However, the actual operation is made clear by considering Fig 1 in conjunction with the seven steps of the OSS algorithm. The implementation of the first five steps is easily seen. However, the execution of step 6 bears further discussion. Note that step 6(a) of the algorithm,  $(\Delta_{k-1})' = \frac{1}{2} \Delta_{k-1}$ , is not explicitly executed since  $(X_{k-1})'$  is obtained without performing any arithmetic, but by replacing  $X_{k-1}$  with the available value of  $X_k$ . Step 6(c) is a null operation since  $(\Delta_k)' = \Delta_k$ . Finally, when overshoot is detected, i.e., when  $Y = 1$ , the switches S1 and S2 are placed in the OSS position. In this way steps 6(b) and 6(e) are executed. Finally,  $(X_k)'$  is obtained by using adder A3 to produce  $(X_k)' = (X_{k-1})' + (\Delta_k)' = X_k + \Delta_k$ . Note that the three adders A1, A2 and A3 are really one time-shared adder. These additional steps of the algorithm place an added requirement on the logic speed. After the completion of a normal cycle of the ADM, extra time is needed to perform one more addition and the various logic operations needed to rearrange the internal values. Usually this can be done in one sampling period.

#### IV. COMPUTER VERIFICATION OF THE ALGORITHM

The Digital-Song Delta Modulator, with and without OSS was simulated on a PDP-3 computer. The minimum step size used ( $\Delta$ ) was normalized to unity. The dynamic range was set to correspond to a ten bit internal arithmetic in an actual hardware implementation. Thus, the signal estimate  $X_k$  was permitted to vary from 0 to  $1024\Delta$ .

The responses of the DM to step functions of different amplitudes, with and without OSS, appear in Fig 6. Figures 6(a) and 6(b) exhibit large overshoots and sustained oscillations (see also Fig 8(b)). They correspond to the sequence  $e_{k-3} = e_{k-2} = 1, e_{k-1} = e_k = -1$ , where  $k-1$  is the sampling time when overshoot occurs. Figures 6(a') and 6(b') are the same waveforms but with overshoot suppression (see also Fig 8(c)). As an example compare Figs 6(a) and 6(a'). Here the maximum peak to peak oscillations are reduced from  $22\Delta$  to  $9\Delta$ . Similar observations can be made for Figs 6(b) and 6(b'). Furthermore, here the settling time to the steady state is reduced from six to three sampling periods. While Fig 6 gives a good indication of the general nature of the improvement due to OSS, a more convincing illustration is depicted in Fig 7 where the discontinuities are much larger. Note that the apparent slow rise times in Fig 7 are due to the compression produced by a scaling factor of 0.1 used in the plotting. In reality Fig 7 rises over a range of  $500\Delta$  in only 13 sampling periods. To achieve the same amplitude, a non-adaptive DM would require over 500 sampling periods.

Briefly, the salient features of the response are as follows. The rise time to reach a given level is the same with or without OSS. Overshoots are suppressed by a minimum of 50%. Recovery times following overshoots are significantly reduced as seen in Fig 7(b). An analysis of the recovery time improvement is given in Section VII. The data plotted in Fig 7 is given in Table 1 for quantitative comparisons. The peak-to-peak amplitude of the steady state response is three times the minimum step size for either scheme. The period of steady state



oscillations is 4 sampling periods without OSS, and 8 sampling periods with OSS. In either case, the peak-to-peak steady state oscillation amplitudes can be made smaller than a grey level in the picture waveform. Thus, constant shade regions will not suffer significant degradation (granularity and contouring).

## V. EXPERIMENTAL INVESTIGATIONS

The experimental test set-up that was used is built around a PDP-8 computer equipped with multichannel A/D and D/A converters. The computer was programmed to perform real time DM processing of the analog signals applied to the A/D converter. Provision was made for implementing an OSS option. The various channels of the D/A converter were programmed to output the DM estimates of the analog inputs, as well as synchronized scanning waveforms to produce a video display of the DM estimates on a monitor.

To display the edge effects of delta modulation in video processing a square wave input was used. The square wave had a repetition rate of 350 Hz and was bandlimited to 10Khz to simulate the rise times of realistic bandlimited waveforms. The DM sampling frequency is 40Khz. The original square wave and the filtered waveforms appear in Fig 8(a). Figure 8(a') is the 100 line raster produced with the filtered waveform of Fig 8(a) as the Z modulation. Note that except for the narrow vertical bright band at the leading edge of each bar, which is due to the ringing introduced by the filter, the brightness of the bar is uniform and the edges are sharply defined.

Figure 8(b) shows the delta modulated version, without OSS, of the filtered waveform in Fig 8(a). The lower waveform in Fig 8(b) is just the delta modulated output waveform bandlimited to  $f_m = 10$  KHz, (the bandwidth of the input to the DM). The raster in Fig 8(b') results when the filtered waveform of Fig 8(b) is applied to the Z-input of the monitor. The deleterious effects of the overshoots and subsequent oscillations on the leading edge are obvious. Note also that the

gradual roll-off at the trailing edge of the waveform causes a lack of sharp contrast at the trailing edge of the bright bars.

Figures 8(c) and 8(c') correspond to Figs 8(b) and 8(b') respectively, when OSS is applied. The improvement in Fig 8(c') over 8(b') is again obvious. In fact, the only difference between Fig 8(a') and Fig 8(c') is the lack of sharp contrast on the trailing edges in Fig 8(c'). This is due to the slower fall-time of the delta modulated waveform than that of the original filtered waveform. The way to improve this is to increase  $\alpha + \beta$  which will lead to faster rise time capabilities of the DM. This is now possible because the overshoot effects on the leading edge are taken care of by the OSS scheme.

As a further and more realistic verification of the viability of OSS in video transmission, the square wave input was replaced by a flying-spot scanner, and a realistic still picture was ADM processed using the PDP-8 computer. A frame consisted a 400 DM bits per line with 170 lines per frame. The minimum DM step-size was adjusted for 64 grey levels/pixel while  $\alpha$  and  $\beta$  were set equal to 1 and 0.5, respectively. In scanning the frame each DM estimate was displayed with a slight spatial overlap with its neighboring estimates. This in effect is an "averaging" procedure which tends to low-pass filter the picture. As such, some of the overshoots are filtered out without the OSS algorithm, which is itself effectively a low-pass filter for overshoots. Averaging has, in general, the added advantage of eliminating much of the "busyness" introduced by the ADM.

The picture chosen for processing has areas of much detail as well as relatively quiet areas. This permits us to display the limitations of the ADM as well as of the OSS algorithm. The results appear in Fig (9). Although averaging has removed some of the overshoots, many bright spots are still observable on the jacket, especially along the edge with the shirt, as well as on the hair, and along the hair and face border in the picture without OSS. The picture with OSS, but

without a threshold (see the threshold discussion on page 8), has no overshoot but all detail is smeared out. Note especially the background area with the trees. This picture is certainly unsatisfactory. Finally, consider the picture with OSS but with a threshold set at 8 grey levels. Here practically all detail is restored and all but a few overshoots along the jacket and shirt edge are removed. Thus, OSS with a threshold appears to perform better for picture transmission than similar schemes without OSS and a threshold. It should be pointed out, however, that although overshoots constitute a lack of fidelity, they may sometimes be desirable. This is due to the subjective edge enhancement overshoots sometimes produce. Thus, the decision to implement OSS or not ultimately depends on the system use requirements.

## VI. ADM STABILITY CONSIDERATIONS

As previously mentioned, there are various trade-offs in choosing  $\alpha$  and  $\beta$ . To obtain a rapid rise-time capability,  $\gamma = \alpha + \beta$ , should be made large. This may result in a step response containing large amplitude overshoots, and oscillations which last many sampling periods unless OSS is used. Furthermore, some values of  $\alpha$  and  $\beta$  give rise to large amplitude oscillations which do not decay and may even become unbounded. This is an "unstable" mode of operation and is undesirable. In this section we derive conditions on  $\alpha$  and  $\beta$  which insure stability. It is then possible to pick  $\alpha$  and  $\beta$  from this stable set to meet rise-time requirements and subsequently use OSS to reduce any resulting overshoot noise.

The permissible regions of operation in the  $\alpha$  vs  $\beta$  plane are now derived. Considering Eq (4) we must have for proper ADM operation

$$0 < \delta = \alpha - \beta < 1 \quad (11a)$$

and

$$1 < \gamma = \alpha + \beta \quad (11b)$$

to permit the step-sizes to decay and increase, respectively. However, not all points in the  $\gamma$  vs  $\delta$  plane bounded by Eq (11) result in stable operation. (Note, it will be more convenient to use  $\gamma$  and  $\delta$  in the discussion that follows).

Stability is now carefully defined in terms of the DM step response. As seen in Fig (10), the step response consists of a sequence of rising steps until overshoot occurs. Following overshoot the response oscillates with a varying number of points above and below the constant input. The exact settling pattern depends on the particular value of the level  $S(t)$  as well as on  $\gamma$  and  $\delta$ . The slowest settling pattern, and hence the one most likely to start diverging in amplitude, occurs when the response oscillates with  $n$  points above and  $n$  points below the constant input. This shall be called an "n-point" pattern. It turns out that the value of  $n$  depends on the particular  $\gamma$  and  $\delta$  utilized. An ADM is thus defined to be stable if the  $n$ -point oscillatory pattern about a constant input decays. This depends on the value of  $\gamma$  and  $\delta$ . Note that Fig (10) shows a stable case with  $n = 3$ . Note too, that we have defined stability in terms of an  $n$ -point pattern because, as mentioned above, that is the case most likely to start diverging and hence provides a bound on stability. That is, if all  $n$ -point patterns for a given  $\gamma$  and  $\delta$  decay, so too will all other patterns.

The stability conditions for an  $n$ -point pattern are now derived. Consider Fig 10 generalized to an  $n$ -point pattern. The step sizes are

$$\Delta (1)$$

$$\Delta (2) = \gamma \Delta (1)$$

.

.

.

$$\Delta (n) = \gamma^{n-1} \Delta (1)$$

$$|\Delta (n+1)| = \delta \gamma^{n-1} \Delta (1) \quad (12)$$

$$|\Delta (n+2)| = \delta \gamma^n \Delta (1)$$

.

.

.

$$|\Delta (2n)| = \delta \gamma^{2n-2} \Delta (1)$$

To insure stability it is necessary to have  $X(1) < X(2n+1)$ . This is equivalent to

$$\Delta (1) + \Delta (2) + \dots + \Delta (n) > |\Delta (n+1)| + \dots + |\Delta (2n)| \quad (13)$$

Substituting Eq (12) into (13), the stability condition is found to be

$$\delta < \gamma^{1-n} \quad (14)$$

Next, it is necessary to ensure the proper pattern of  $n$ , and only  $n$ , points above  $S(t)$  per half cycle of the oscillations. To have  $n$  points above  $S(t)$ ,  $X(n) < X(2n)$ . This is equivalent to

$$\Delta (n) > |\Delta (n+1)| + \dots + |\Delta (2n-1)| \quad (15)$$

Substituting Eq (12) into (15), we obtain the pattern condition

$$\delta < \frac{1 - \gamma}{1 - \gamma^{n-1}} \quad (16)$$

To have less than  $n + 1$  points above  $S(t)$ ,  $X(2n + 1) < X(n)$ . This is equivalent to

$$\Delta(n) < |\Delta(n + 1)| + \cdots + |\Delta(2n)| \quad (17)$$

Substituting Eq (12) into (17), we obtain

$$\frac{1 - \gamma}{1 - \gamma^n} < \delta \quad (18)$$

Finally, combining Eqs (14), (16), and (18) we obtain the conditions for a stable  $n$ -point pattern

$$\left( \frac{1 - \gamma}{1 - \gamma^n} \right) < \delta < \gamma^{1-n} \quad (19a)$$

and

$$\left( \frac{1 - \gamma}{1 - \gamma^n} \right) < \delta < \left( \frac{1 - \gamma}{1 - \gamma^{n-1}} \right) \quad (19b)$$

Points in the  $\gamma - \delta$  plane bounded by Eq (19b) will produce  $n$ -point pattern oscillations. However, only those points lying in the region bounded by Eq (19a) will be both  $n$ -point as well as stable.

Figure (11) is a plot of the stability regions for  $n = 2$ , and 3. Note that the lower bound on  $\delta$  for  $n = 2$  is the "pattern" upper bound for  $n = 3$ . This is true in general for any  $n$ . Note too, that the only stable pattern in the overlap region of say  $n = 2$  and 3 is  $n = 2$ .

In general, any overlap region will be stable for one, and only one,  $n$ -point pattern.

In conclusion, the results we give permit the selection of  $\gamma$  and  $\delta$  to meet performance specifications, such as rise time requirements, and at the same time guarantee stable operation. We conjecture, without proof, that some unstable  $(\gamma, \delta)$  pairs may operate in a stable fashion when used with OSS. It can in fact be shown, for example, that this is the case for the otherwise unstable pairs  $(2, 0.5)$  and  $(3, 0.5)$ . On the other hand, stable  $(\gamma, \delta)$  will not become unstable when used in conjunction with OSS. Hence, our results guarantee the existence of large stable  $\gamma$ 's, and our OSS scheme permits the minimization of resulting overshoots. Once a stable  $(\gamma, \delta)$  is chosen,  $\alpha = (\gamma + \delta)/2$  and  $\beta = (\gamma - \delta)/2$ . Finally, it is generally desirable to make  $n$  small, namely  $n = 2$ . This results in oscillations with the smallest possible period producing a minimum amount of in-band oscillatory granular noise. Hence, this facilitates the post - ADM low-pass filtering of the noise.

## VII SETTLING TIME CALCULATIONS

The slope overload response of an ADM, as occurs when tracking a step input, is terminated with the occurrence of an overshoot. This is followed by oscillations about the constant level that decrease in amplitude until a small steady state pattern is reached. We define the "settling time" as the time interval between the overshoot and the beginning of the final steady state pattern. Since the oscillations constitute undesirable granular noise, the settling time should be as small as possible. In this section we calculate bounds for the settling time. Consideration is given to the ADM response when operating with or without the OSS algorithm. The quantitative results verify the desirable and significant reduction in the settling time with the use of OSS.

### A. A Settling Time Upper Bound Without OSS

When the first overshoot occurs in a step response at time  $k$  with a step-size  $\Delta_k$  as shown in Fig (12a) the input  $S(t)$  may be anywhere between  $X_{k-1}$  and  $X_{k-1} + \Delta_k$ . The longest settling time occurs when  $S(t)$  falls in the middle of  $\Delta_k$  so that the oscillations decay in a symmetric  $n$ -point pattern. As shown in Section VI, the pattern is determined by the particular  $(\gamma, \delta)$  used. Since a 2-point pattern provides, in general, the shortest settling time and produces the smallest percentage of in-band granular noise, we shall consider only this particular practical case.

In Fig (12a) the step size which causes the overshoot is  $\Delta_k$ , while the step-size magnitudes at  $k+1, k+2, \dots$  are given by

$$|\Delta_{k+1}| = \delta \Delta_k$$

$$|\Delta_{k+2}| = \gamma \delta \Delta_k$$

$$|\Delta_{k+3}| = \gamma \delta^2 \Delta_k$$

$$|\Delta_{k+4}| = \gamma^2 \delta^2 \Delta_k$$

Thus, the emerging pattern is given by

$$|\Delta_{k+N}| = \begin{cases} \gamma^{\frac{N-1}{2}} \delta^{\frac{N+1}{2}} \Delta_k, & N \text{ odd} \\ \gamma^{\frac{N}{2}} \delta^{\frac{N}{2}} \Delta_k, & N \text{ even} \end{cases} \quad (20a)$$

$$(20b)$$



Next, the settling time  $N$  is defined as the minimum time required for  $\Delta_{k+N} = \Delta$ , i.e., the minimum number of sampling periods that elapse, following the initial overshoot, until the final steady state oscillatory pattern is reached. Recalling the conditions,  $\gamma > 1$  and  $0 < \delta < 1$ , it is apparent from Fig 12(a) that the first minimum step size occurs at  $\Delta_{k+N} = \Delta$  when  $N$  is odd. Thus, using Eq (20a) we obtain the following upper bound on the settling time when  $(\gamma, \delta)$  produces a 2-point pattern:

$$N_{\max} = \frac{2 \ln \left[ \frac{\Delta}{\Delta_k} \sqrt{\frac{\gamma}{\delta}} \right]}{\ln(\gamma \delta)} \quad (21)$$

The reduction in  $N_{\max}$  when OSS is utilized is calculated in the following section.

#### B. A Settling Time Upper Bound With OSS

The worst case settling time with OSS, corresponding to Fig 12(a), is shown in Fig 12(b). In this case OSS is implemented every third sampling instant, and the amplitude prior to suppression is indicated with dashed lines. Here the step-size at time  $k$  is  $\Delta'_k = \frac{1}{2} \Delta_k$  as a result of the suppression algorithm. Further,

$$\begin{aligned} |\Delta'_{k+1}| &= \frac{1}{2} \Delta_k \\ |\Delta'_{k+2}| &= \frac{1}{2} \delta \Delta_k \\ |\Delta'_{k+3}| &= |\Delta'_{k+4}| = \frac{1}{4} \gamma \delta \Delta_k \\ |\Delta'_{k+5}| &= \frac{1}{4} \gamma \delta^2 \Delta_k \\ |\Delta'_{k+6}| &= \frac{1}{8} \gamma^2 \delta^2 \Delta_k \end{aligned}$$

The pattern that emerges is given by

$$\left| \Delta'_{k+3m-1} \right| = \frac{1}{2^m} \delta^m \gamma^{m-1} \Delta_k, \quad m = 1, 2, 3, \dots \quad (22a)$$

and

$$\left| \Delta'_{k+3m} \right| = \left| \Delta'_{k+3m+1} \right| = \frac{1}{2^{m+1}} (\gamma \delta)^m \Delta_k, \quad m = 0, 1, 2, 3, \dots \quad (22b)$$

In considering the settling time to the steady state here, it is not clear whether  $\Delta_{k+N} = \Delta$  for  $N$  odd or even. This depends on the particular  $(\gamma, \delta)$  used. However, for  $(\gamma, \delta)$  chosen from the stability region for  $n = 2$  in Fig (11) it was found that  $N = 3m$  gives a slightly pessimistic estimate of the settling time. We thus use Eq (22b) to find the maximum settling time for a 2-point pattern with OSS

$$N_{\max} \mid_{\text{OSS}} = 3m = \frac{3 \ln \left[ \frac{2\Delta}{\Delta_k} \right]}{\ln \left[ \frac{\gamma \delta}{2} \right]} \quad (23)$$

$N_{\max} \mid_{\text{OSS}}$  is plotted in Fig (13) as a function of  $\Delta_k$  with  $(\gamma, \delta)$  as parameters. It is noted that  $N_{\max} \mid_{\text{OSS}}$  increases with  $\gamma$ . Thus, although large  $\gamma$  is desirable to reduce slope overload noise, it is undesirable because of the resulting increases in granular noise. However, the latter is reduced using OSS as seen by comparing Eqs (21) and (23). The percentage improvement in settling time due to OSS is plotted in Fig (14). It is seen that the improvement increases with  $\gamma$ . This is desirable since the settling time increases with  $\gamma$  in general. Finally, we see that there is a better than 50% improvement for the practical case  $\gamma = 1.5$  and  $\delta = 0.5$ .

### VIII Rise Time Formulas and Truncation Errors

The ADM, when operating in slope overload, can be modeled as a nonlinear digital low-pass filter. The shortest risetimes that can be handled by this model without slope-overload obviously depend on its parameters  $\Delta$  and  $\gamma$ . The speed with which the step-size increases is a function of  $\gamma$ , higher values for  $\gamma$  will ensure shorter rise times and therefore are to be preferred in video transmission. Large values of  $\gamma$ , however, as already indicated, will lead to large overshoots and instabilities unless some OSS scheme is used.

In what follows, in order to illustrate the complexity of the mathematical models needed for investigating this problem, and as a first step towards its solution, we derive a formula for the minimum rise time obtainable under given initial conditions at the beginning of an input discontinuity. For this purpose, the rise time  $t_r$  (for brevity we will use in what follows the simplifying notation  $t_r = i$ ) is defined by the number of sampling instances needed in a given ADM (given  $\gamma, \Delta$ ) in order to attain a specified voltage level  $V$ . Namely, we look for a relation of the form

$$i = f_1(V \mid \gamma, V(0), \Delta_1) \quad (24)$$

where  $V(0)$  is the DM voltage level at the start of the discontinuity and  $\Delta_1$  is the first step size.

It is shown below that the main difficulty in deriving an explicit formula for Eq (24) is due to the truncation errors inherent in a digital implementation where  $V(i)$  and  $\Delta(i)$  are constrained to be integer valued while  $\gamma$  may be noninteger valued. These errors become too large to be neglected as  $i$  increases.

To derive Eq (24) we first derive an expression for

$$V = f_2(i | \gamma, V(0), \Delta_1) \quad (25)$$

assuming no truncation errors (infinite arithmetic capabilities). Eq (25) is obtained by summing  $V(0)$  and the increasing step sizes

$$V(i | \gamma, V(0), \Delta_1 > \Delta, \text{no truncation}) = V(0) + \Delta_1 + \gamma \Delta_1 + \dots + \gamma^{i-1} \Delta_1 \quad (26)$$

which yields

$$V(i | \gamma, V(0), \Delta_1 > \Delta, \text{no truncation}) = V(0) + \Delta_1 \left( \frac{1-\gamma^i}{1-\gamma} \right), i = 1, 2, \dots \quad (27)$$

Note that Eq(27) must be modified if  $\Delta_1$  is the minimum step size, i.e.,  $\Delta_1 = \Delta$ . For this case

$$V(i | \gamma, V(0), \Delta_1 = \Delta, \text{no truncation}) = V(0) + \Delta + M + \gamma M + \dots + \gamma^{i-2} M \quad (28)$$

which yields

$$V(i | \gamma, V(0), \Delta_1 = \Delta, \text{no truncation}) = V(0) + \Delta + M \left( \frac{1-\gamma^{i-1}}{1-\gamma} \right), i=2,3,\dots \quad (29)$$

Under the no-truncation assumption, Eq (24) is obtained by straightforward algebra from Eqs (27) and (29), yielding

$$i = \begin{cases} \ln \left[ 1 + \frac{(\gamma - 1)(V - V(0))}{\Delta_1} \right] / \ln(\gamma), & \Delta_1 > \Delta \\ \ln \left\{ \gamma \left[ 1 + \frac{(\gamma - 1)}{M} (V - V(0) - \Delta) \right] \right\} / \ln(\gamma), & \Delta_1 = \Delta \end{cases} \quad (30a)$$

$$(30b)$$

These values obtained for  $V(i)$  in Eqs (27) and (29) and for  $i(V)$  in Eq (30) are actually upper and lower bounds, respectively, for the true values in a digital implementation because the actual step sizes  $\Delta_k$  given by Eq (4) are obtained with all fractional parts truncated. Table (2) shows the discrepancy in voltage between the processes with and without truncation for  $\gamma = 1.5$ . Let  $V_T(i)$  and  $V_{NT}(i)$  be the values of  $V(i)$  with and without considering the truncation error, respectively. Also, let  $\epsilon_{cum}(i)$  be the cumulative truncation error in the voltage  $V(i)$ . We then have

$$V_T(i) = V_{NT}(i) - \epsilon_{cum}(i) \quad (31)$$

Next,  $\epsilon_{cum}(i)$  is computed. It should be noted that it results from the local round-off error,  $r$ , in the step size at each sampling instant. The analysis is complicated by the fact that  $r$  is a random variable, with unknown statistics, assuming different values in the range,  $0 \leq r < 1$ , at each sampling instant. To make this problem tractable, we assume that  $r$  is always the same and occurs only at every other sampling instant. These assumptions were motivated by the examination of many typical error patterns and are ultimately justified by the accuracy of the results given. It should be pointed out here that  $r$  is a scaling factor in our final formula, Eq (35), and that we shall give a rule to pick the proper  $r$  to obtain accurate results for a given value of  $\gamma$ . Finally, using the two assumptions mentioned above, the cumulative truncation error in  $V$  at time  $n$ , where the first step size round-off error occurs at  $n = 1$ , is given by

$$\epsilon_{cum}(n) = \sum_{k=1}^n \epsilon(k) \quad (32)$$

Note that  $\epsilon_{(k)}$  is the local truncation error in the step size resulting from the previous and present round-off errors. It is given by the following recursion relation

$$\epsilon_{(k+1)} = \gamma \epsilon_{(k)} + \frac{r}{2} \left[ 1 + (-1)^k \right] \quad (33)$$

subject to the initial condition  $\epsilon_{(0)} = 0$ . Equation (33) can be easily solved using Z transform methods, yielding

$$\epsilon_{(k)} = \frac{r\gamma^{k+1}}{\gamma^2 - 1} - \frac{r(-1)^k}{2(\gamma+1)} + \frac{r}{2(1-\gamma)} \quad (34)$$

Thus, the cumulative truncation error is given by

$$\begin{aligned} \epsilon_{\text{cum}}(n) = \sum_{k=1}^n \epsilon_{(k)} &= \frac{r\gamma(\gamma - \gamma^{n+1})}{(\gamma^2 - 1)(1-\gamma)} - \frac{r[(-1)^n - 1]}{4(\gamma + 1)} \\ &+ \frac{r n}{2(1-\gamma)}, \quad n = 1, 2, 3, \dots \end{aligned} \quad (35)$$

It is now possible to find  $V_T(i | \gamma, V(0), \Delta_1)$  by combining Eqs (27), (29), (31) and (35). Considering Eq (27), note that if  $\Delta_1 > \Delta$ , then truncation may start occurring at  $i \geq 2$ . Thus, we introduce the index shift  $i = n - 1$ ,  $n \geq 2$ ,  $i \geq 1$  in Eq (35). Combining the result with Eqs (27) and (31) we obtain

$$V_T(i | \gamma, V(0), \Delta_1 > \Delta) = \begin{cases} V(0) + \Delta_1, & i = 1 \\ V(0) + \Delta_1 \left( \frac{1-\gamma^i}{1-\gamma} \right) - \epsilon_{\text{cum}}(i | \Delta_1 > \Delta), & i = 2, 3, \dots \end{cases} \quad (36)$$

where  $\epsilon_{\text{cum}}(i \mid \Delta_1 > \Delta) = \epsilon_{\text{cum}}(n) \delta_{n, i-1}$  and  $\delta_{i,j}$  is the Kronecker delta. Similarly in Eq (29) truncation may start occurring at  $i \geq 3$ . Hence, we introduce the index shift  $i = n - 2$ ,  $n \geq 3$ ,  $i \geq 1$  in Eq (31), obtaining

$$V_T(i \mid \gamma, V(0), \Delta_1 = \Delta) = \begin{cases} V(0) + \Delta & , i = 1 \\ V(0) + \Delta + M & , i = 2 \\ V(0) + \Delta + M \left( \frac{1 - \gamma^{i-1}}{1 - \gamma} \right) - \epsilon_{\text{cum}}(i \mid \Delta_1 = \Delta) & , i = 3, 4, 5, \dots \end{cases} \quad (37)$$

where  $\epsilon_{\text{cum}}(i \mid \Delta_1 = \Delta) = \epsilon_{\text{cum}}(n) \delta_{n, i-2}$ .

To make Eqs (36) and (37) meaningful and accurate, we present the following procedure to find the proper  $r$  for a given  $\gamma$ . Let

$\epsilon_{\text{cum}}(i_{\text{max}} \mid \Delta_1) \mid_{\text{actual}}$  be the actual value of the truncation when  $i_{\text{max}}$  corresponds to the maximum voltage excursion to be considered.

Note, this value may always be obtained by simulation of the DM with truncation. Next, for any value  $0 < r < 1$ , say  $r = 0.5$ , find

$\epsilon_{\text{cum}}(i_{\text{max}} \mid \Delta_1) \mid_{r=0.5}$  as defined in Eqs (36) and (37). Then the value of  $r$  that gives accurate results in Eqs (36) and (37) for all  $i \leq i_{\text{max}}$  is given by

$$r = 0.5 \frac{\epsilon_{\text{cum}}(i_{\text{max}} \mid \Delta_1) \mid_{\text{actual}}}{\epsilon_{\text{cum}}(i_{\text{max}} \mid \Delta_1) \mid_{r=0.5}} \quad (38)$$

The derivation of this formula should be apparent by noting that  $r$  is actually a scaling factor in Eq (35).

The above equations and procedures, although heuristic and approximate in the sense that they provide a deterministic solution for a random process, have been tested and shown to be very accurate

for all practical examples. We give results for  $r$  as a function of  $\gamma$  ( $1 \leq \gamma \leq 4$ ) in Fig (15) and Table (3). Here it is assumed that  $V(i_{\max})$  corresponds to the maximum voltage excursion possible, without overflow, in a digital implementation with a ten bit storage register for  $V(i)$  and  $\Delta(i)$ . It is also assumed that  $\Delta = 1$  and  $M = 2$  for all values of  $\gamma$  except  $\gamma = 1.25$  where  $M = 4$ . For the values of  $r$  in Table (3) it was found that Eq (37) gave accurate results for  $V(i)$  when rounded off to the nearest integer. The worst cases occurred for  $\gamma$  equal to 1.25 and 1.5 where the maximum discrepancies were 6 and 2 respectively. This can be compared to the results given by Eq (29) where the maximum discrepancy for  $\gamma$  equal to 1.25 and 1.5 is greater than 400 and 200 respectively. It was further found that Eq (37) gives values whose accuracy improves with increasing  $\gamma$ . In fact, for  $\gamma > 1.75$  the given results for  $V(i)$  are exact.

As mentioned earlier, Eq (30) gives a lower bound for  $i$  in Eq (24) since it does not take into consideration the effects of truncation. A more accurate formulation for Eq (24) results if Eqs (36) and (37) are solved for  $i$ . However, an exact closed form solution is not possible since Eqs (36) and (37) are transcendental in  $i$ . A converging iterative approach may be used where we solve for  $i$  by finding  $\gamma^i$  and taking logarithms. This results in the following recursion formula for  $i$ :



$$i_{n+1} = \begin{cases} \ln \left\{ \frac{\gamma (\gamma^2 - 1) \left[ (V(i) - V(0) - \Delta)(1 - \gamma) - (r + M) + \frac{r\gamma^2}{\gamma^2 - 1} + \frac{r}{2} i_n \right]}{r\gamma - M(\gamma^2 - 1)} \right\} / \ln(\gamma) & , \Delta_1 = \Delta \\ \ln \left\{ \frac{(\gamma^2 - 1) \left[ (V(i) - V(0))(1 - \gamma) - \Delta_1 + \frac{r}{2} i_n \right] + \frac{r}{2} (\gamma^2 + 1)}{[r\gamma - \Delta_1 (\gamma^2 - 1)]} \right\} / \ln(\gamma) & , \Delta_1 > \Delta \end{cases} \quad (39)$$

The questions of convergence and accuracy of Eq (39) were investigated for the values of  $\gamma$  and  $r$  in Table (3). For  $i_0 = 0$ , it was found that convergence occurs for all  $\gamma, r$  and  $V(i) \leq 1000$  by the second iteration. Furthermore, when rounded to the nearest integer,  $i_2$  is found to be accurate for all of the above values of  $r, \gamma$ , and  $V(i)$ . Table (4) illustrates the convergence by  $i_2$  for  $\gamma = 1.5$ , and comparison with Table (2) shows the accuracy of  $i_2$  for this case. The general trend found in investigating Eq (39) was that convergence is fairly insensitive to variations in  $r$  but improves very rapidly with increasing  $\gamma$  and  $V$ . Accuracy is found to be perfect when  $r$  is chosen according to Eq (38). On the other hand, although it is approximate, Eq (30) is preferable because it provides a much simpler closed form solution for  $i$ . It was found that the accuracy of Eq (30) improves with increasing  $\gamma$  and degrades with increasing  $V(i)$ . However, for  $\gamma \geq 1.5$  and  $V(i) \leq 1000$  Eq (30) deviates from the correct value of  $i$  by at most unity. Thus, for these cases Eq (50) is to be preferred over Eq (39). The utility of Eq (39) is hence for cases where  $1 < \gamma < 1.5$  and  $V(i) > 1000$ . Furthermore,  $\epsilon_{cum}(i)$  is certainly required for accurate calculations of  $V_T(i)$ .

In conclusion, we have derived a deterministic approach to handling the problem of random truncation error is analyzing the slope overload operation of a digitally implemented ADM. The rise time formulas

derived above in Eq (30) or (39) are important for they specify the rise time limitations of an ADM with a given  $\gamma$ . Considering the ADM in slope overload as a nonlinear low-pass filter, Eqs (30) or (39) in effect specify its bandwidth as a function of  $\gamma$ . Thus, in designing an ADM to handle a video signal with a given bandwidth, Eqs (30) or (39) specify the minimum  $\gamma$  needed to prevent slope overload noise. It should, however, be pointed out that there are other trade-offs in choosing  $\gamma$  and  $\delta$  such as granular noise and, in particular, overshoots and oscillations. The latter may be minimized by the use of the OSS algorithm. However, the exact trade-off must be determined experimentally through subjective evaluations of resulting picture quality.

## IX. CONCLUSIONS

An overshoot suppression algorithm has been proposed and verified by computer simulations using real and computer generated video test patterns. It has been shown that the scheme significantly improves the transient behavior of digital ADM channels that handle video waveforms.

The main advantages of the proposed algorithm are:

(a) It can be easily utilized in optimizing digital ADM channels that can be described by a closed form mathematical formulation and in particular in the Adaptive-Song-Delta Modulator.

(b) The scheme has rather modest requirements for hardware implementation.

(c) It allows for flexible trade-offs between slope-overload and overshoot noise. Therefore, the addition of the OSS algorithm significantly improves the performance of a digital DM channel for picture transmission.

Upper bounds for the settling time, with and without OSS, are also derived showing a significant improvement with the suppression scheme. Stability conditions are given to enable the selection of proper DM parameters when optimizing the trade-off between slope-overload and overshoot noise. To further facilitate the trade-off a rise-time versus voltage level formula is derived taking into consideration truncation errors. This formula gives accurate results for all the practical cases tested.

## X. ACKNOWLEDGMENT

The authors wish to gratefully acknowledge the assistance and many useful discussions with their colleagues Prof W. Rosenberg, and Messrs. N. Scheinberg, M. Steckman and C. Strassberg.

XI. REFERENCES

1. C. L. Song, J. Garodnick, and D. L. Schilling,  
"A Variable-Step-Size Robust Delta Modulator", IEEE  
Trans. Comm. Tech., Vol COM-19, Dec. 1971, pp 1033-1044.
2. T. Oshima, T. Ishiguro, "Reduction of Edge Busyness in  
Delta Modulation", to be published.
3. M. R. Winkler, "Pictorial Transmission with HIDM", in  
1965 IEEE Int. Conv. Rec., pt. 1, pp. 285-291.
4. N. S. Jayant, "Adaptive Delta Modulation with a One-Bit  
Memory", B.S.T.J., 49, No. 3, March 1970, pp 321-342.
5. J. E. Abate, "Linear and Adaptive Delta Modulation", Proc.  
of the IEEE, Vol.55, No. 3, March 1967, pp 298-308.
6. M. Oliver, "An Adaptive Delta Modulator with Overshoot  
Suppression for Video Signals", IEEE Trans. Comm. Tech.,  
Vol. COM-21, March 1973, pp 243-247.
7. L. Weiss, I. Paz, D. L. Schilling, "Overshoot Suppression  
In Adaptive Delta Modulator Links for Video Transmission",  
Proc. of the 1973 IEEE National Telecommunications Conference,  
Atlanta, Georgia, Nov. 1973, pp 6d1-6d6.
8. C. L. Song, J. Garodnick, D. L. Schilling, "An Adaptive Delta  
Modulator for Speech and Video Processing", Proc. of the  
1972 IEEE International Conference on Commun., Philadelphia,  
Pa., June 1972, pp 21.30-21.31.

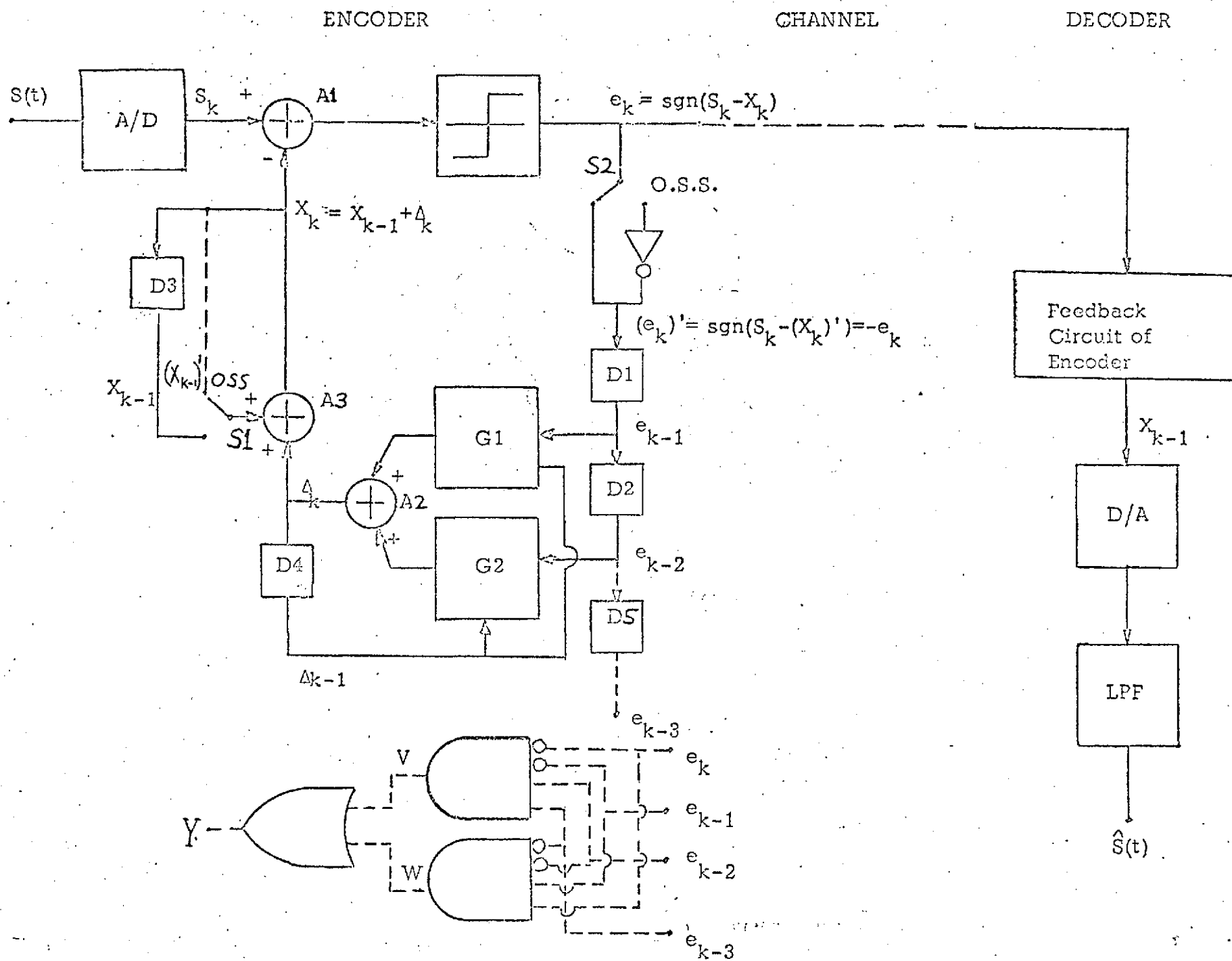


Fig.1 Song-Adaptive-Delta Modulator with provisions for Overshoot Suppression. (Dashed branches are for Overshoot Suppression.)

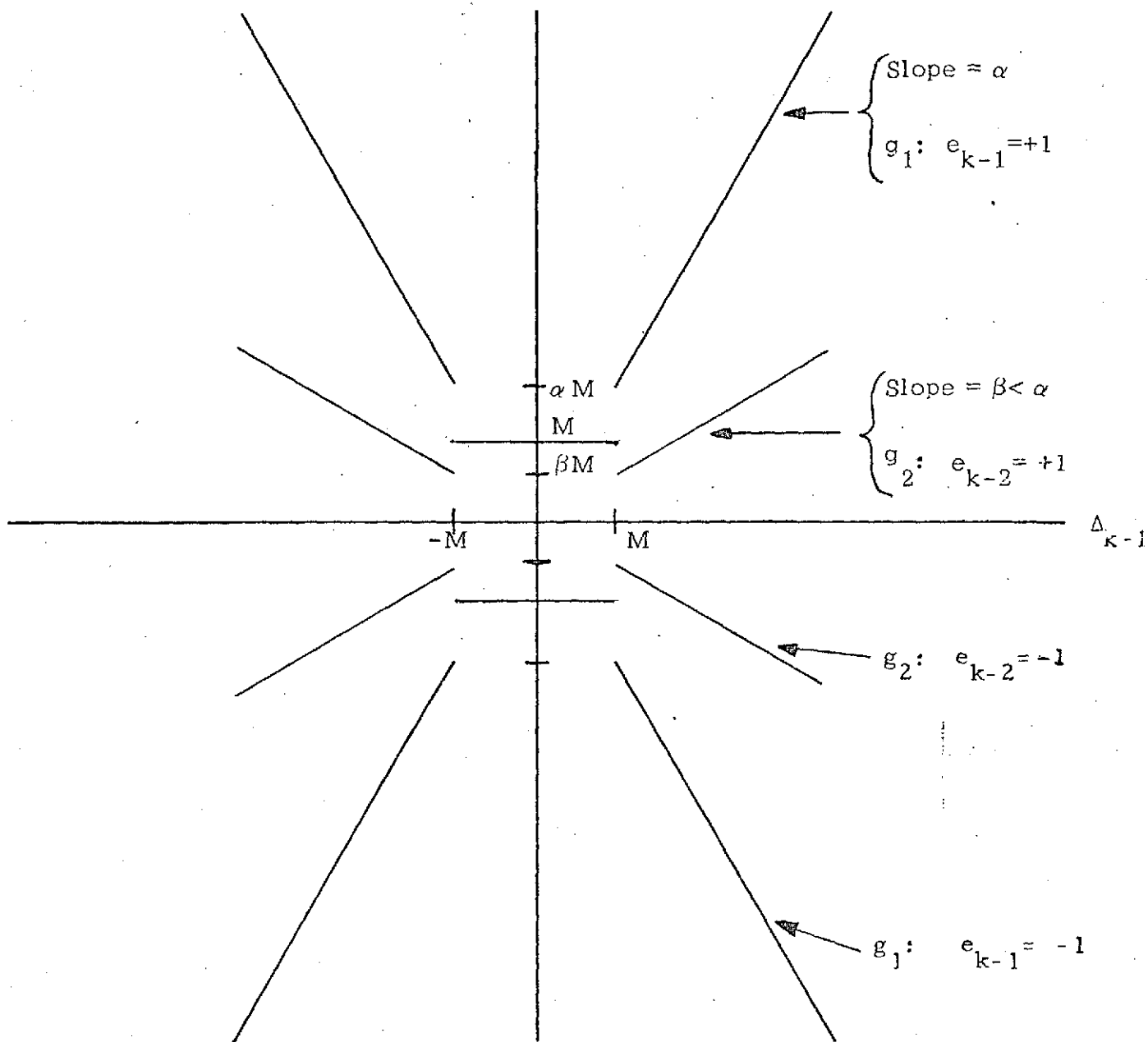


Figure 2 Step-size  $g_1, g_2$  Characteristics.

NOTE: The following conditions are needed for step size adaptability:

- (a)  $\alpha + \beta = \gamma > 1$
- (b)  $0 < \alpha - \beta = \delta < 1$

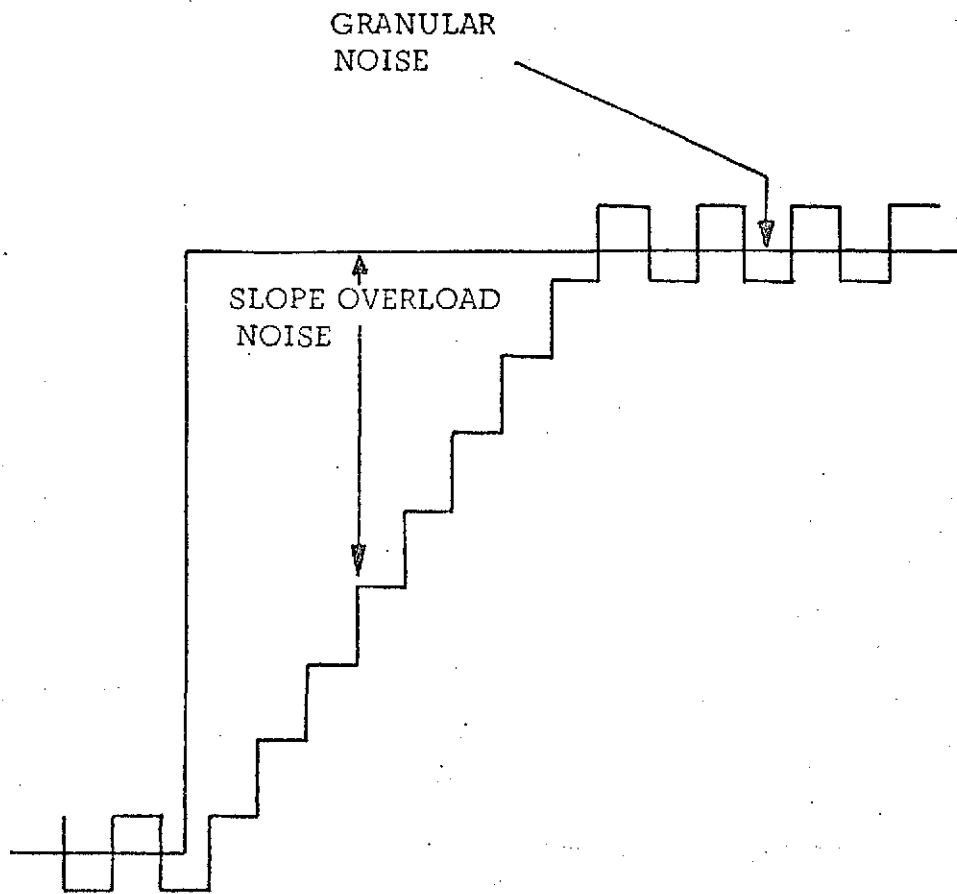


Figure 3(a) Linear Delta Modulator step response exhibiting Slope Overload and Granular Noises.

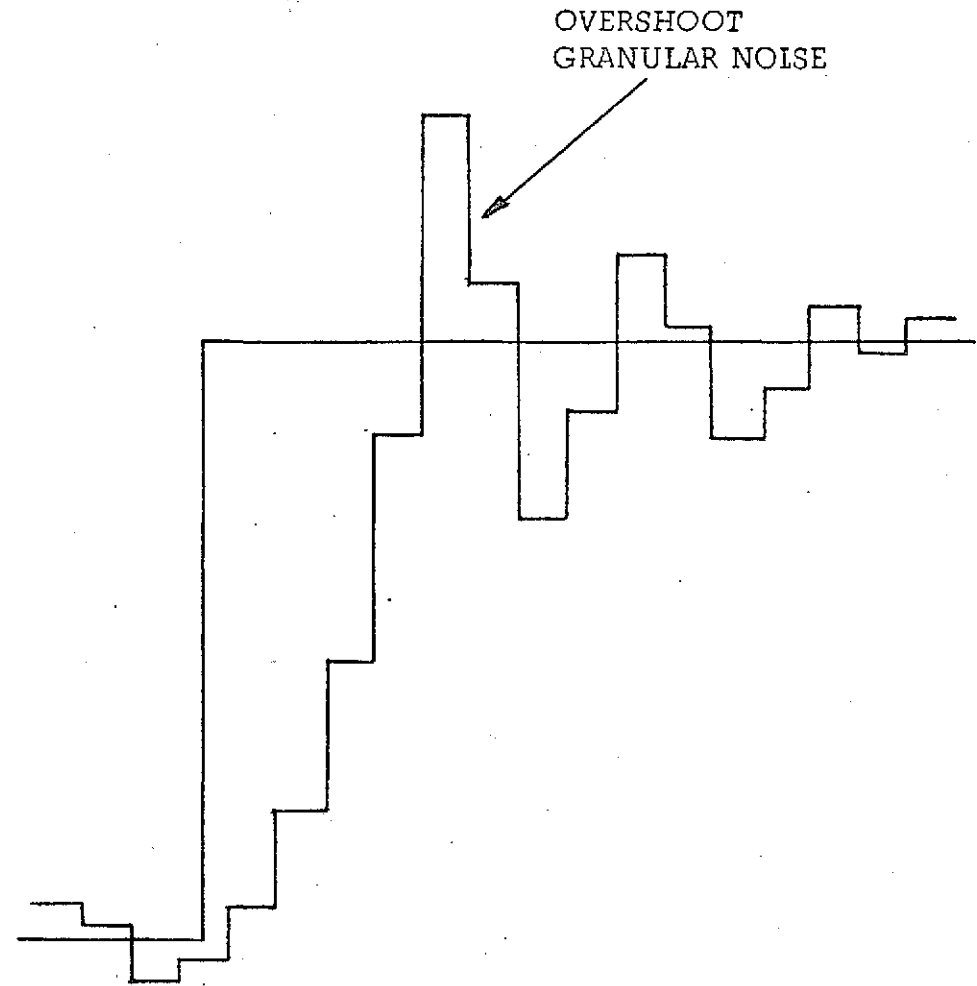


Figure 3(b) Adaptive Delta Modulator step response exhibiting overshoots and oscillations.



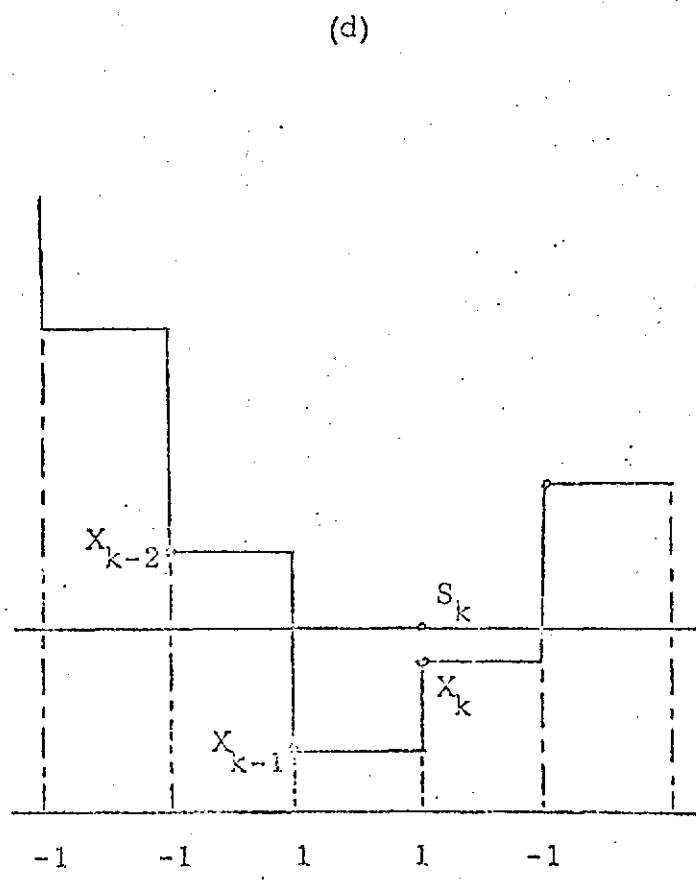
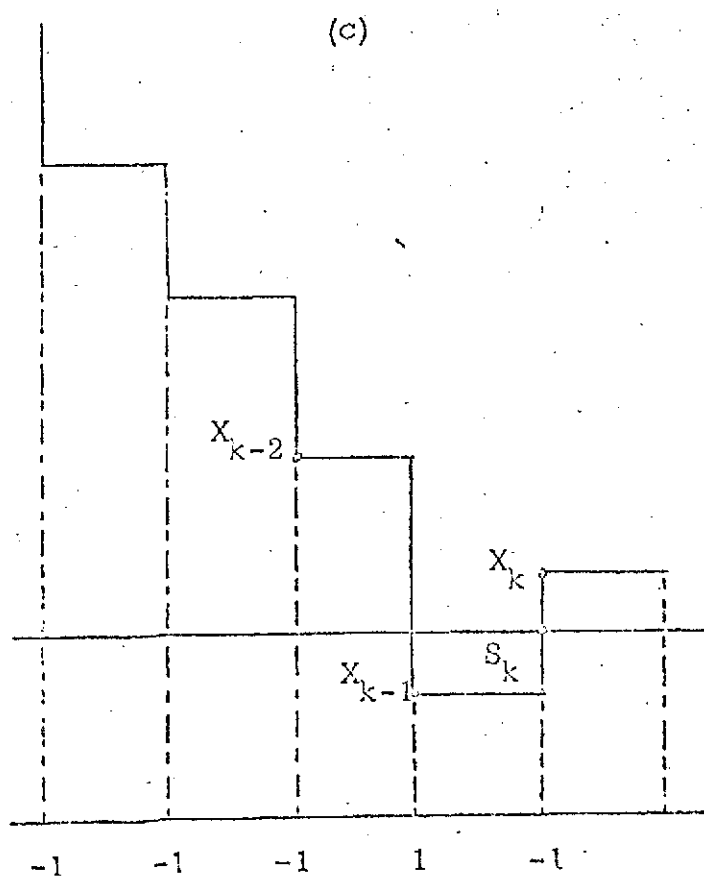
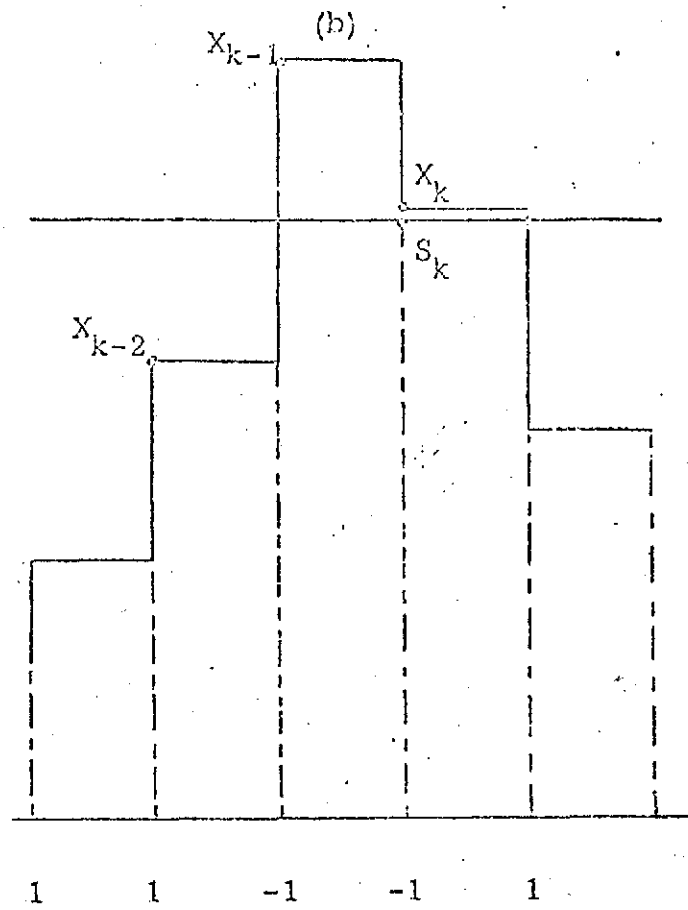
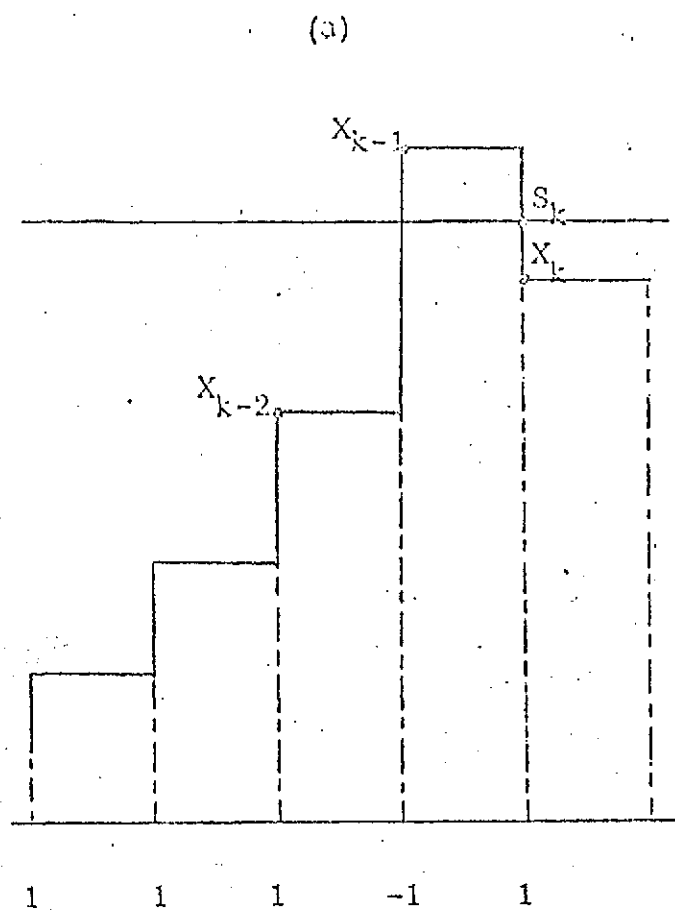


Fig. 4 The four cases of overshoot in the step response of the Song-Adaptive-Delta Modulator.

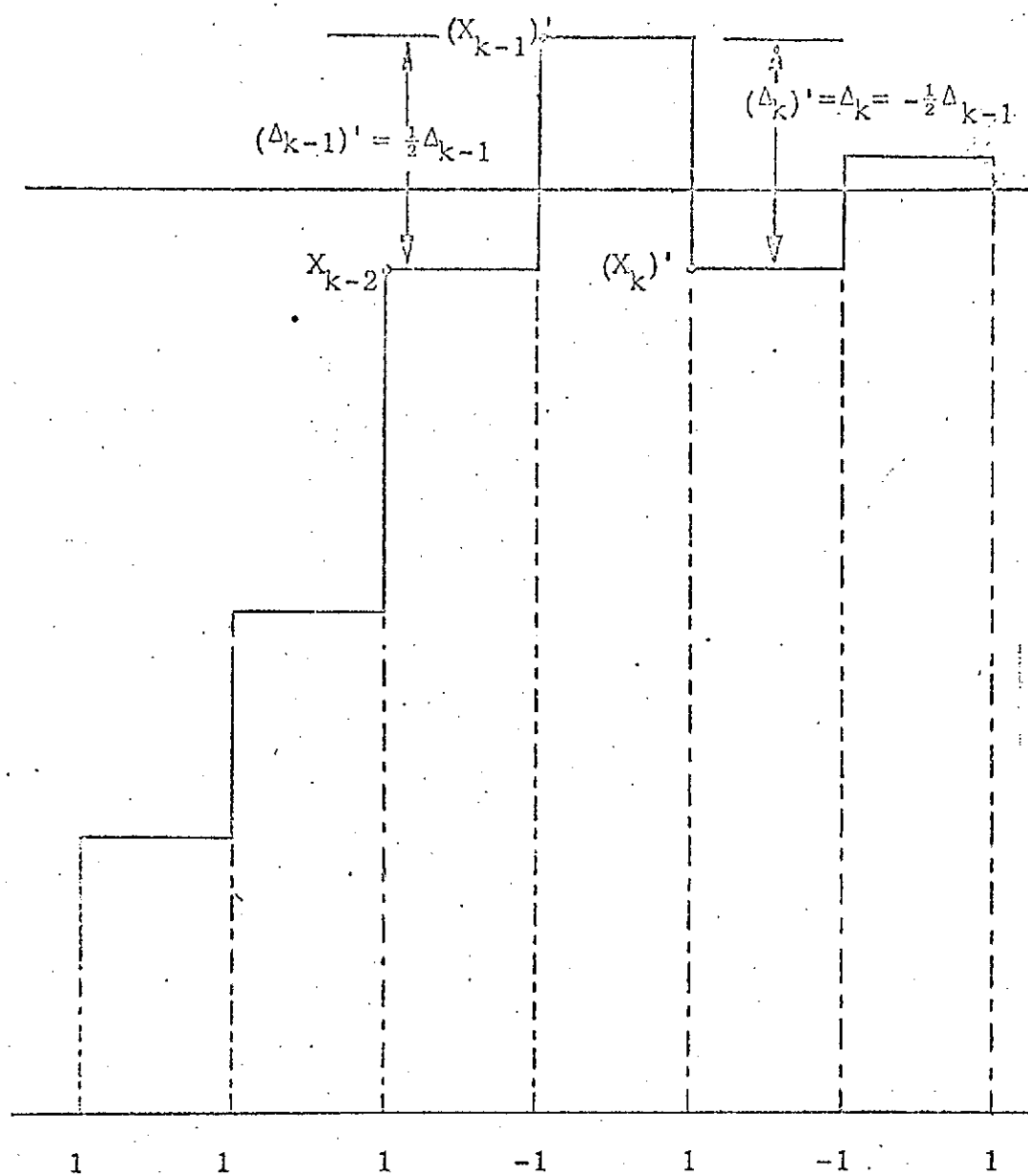


Fig. 5. Figure 4b after Overshoot Suppression.

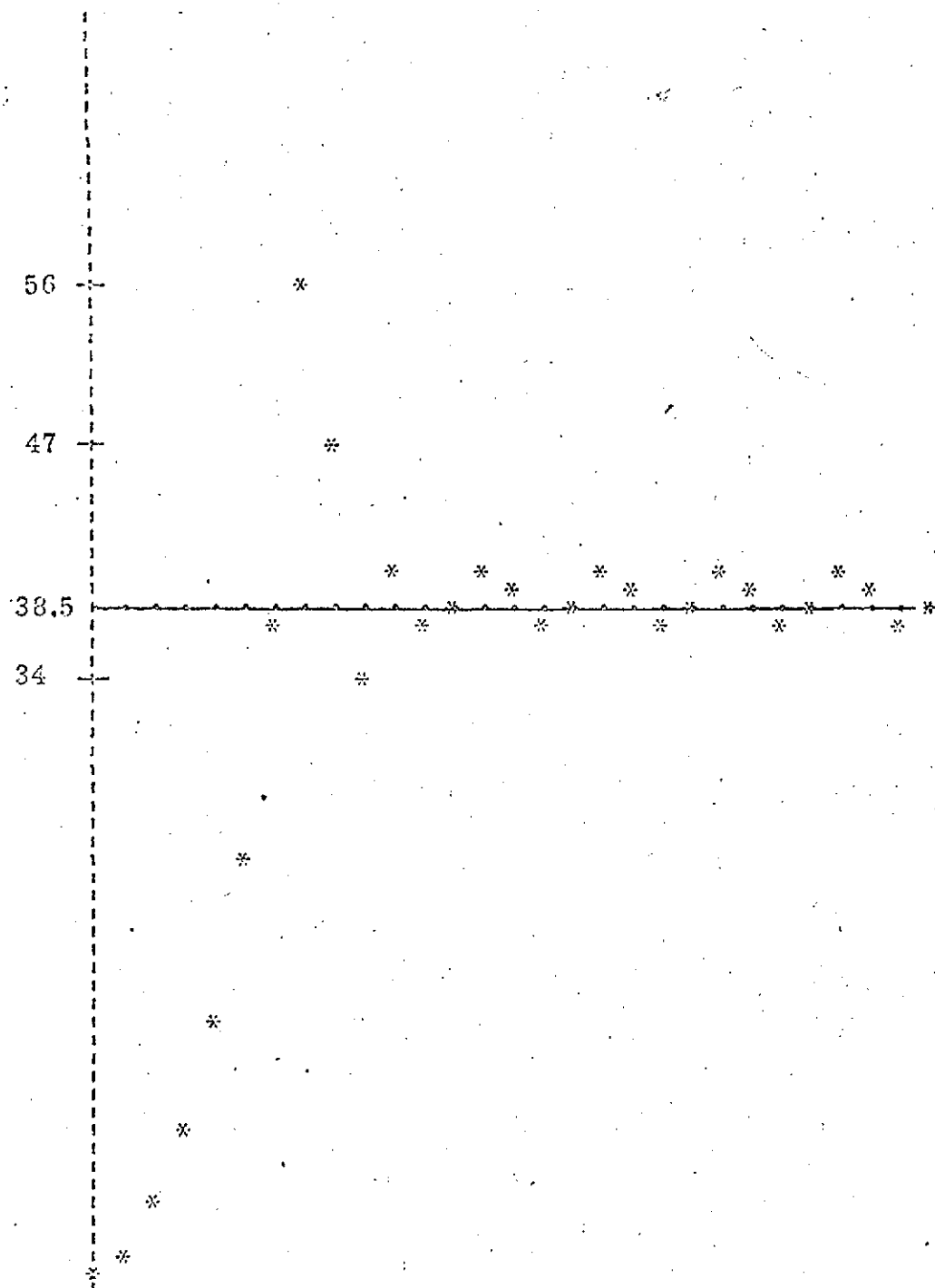


Fig. 6 (a) The step response of a Song-Adaptive Delta Modulator without Overshoot Suppression.

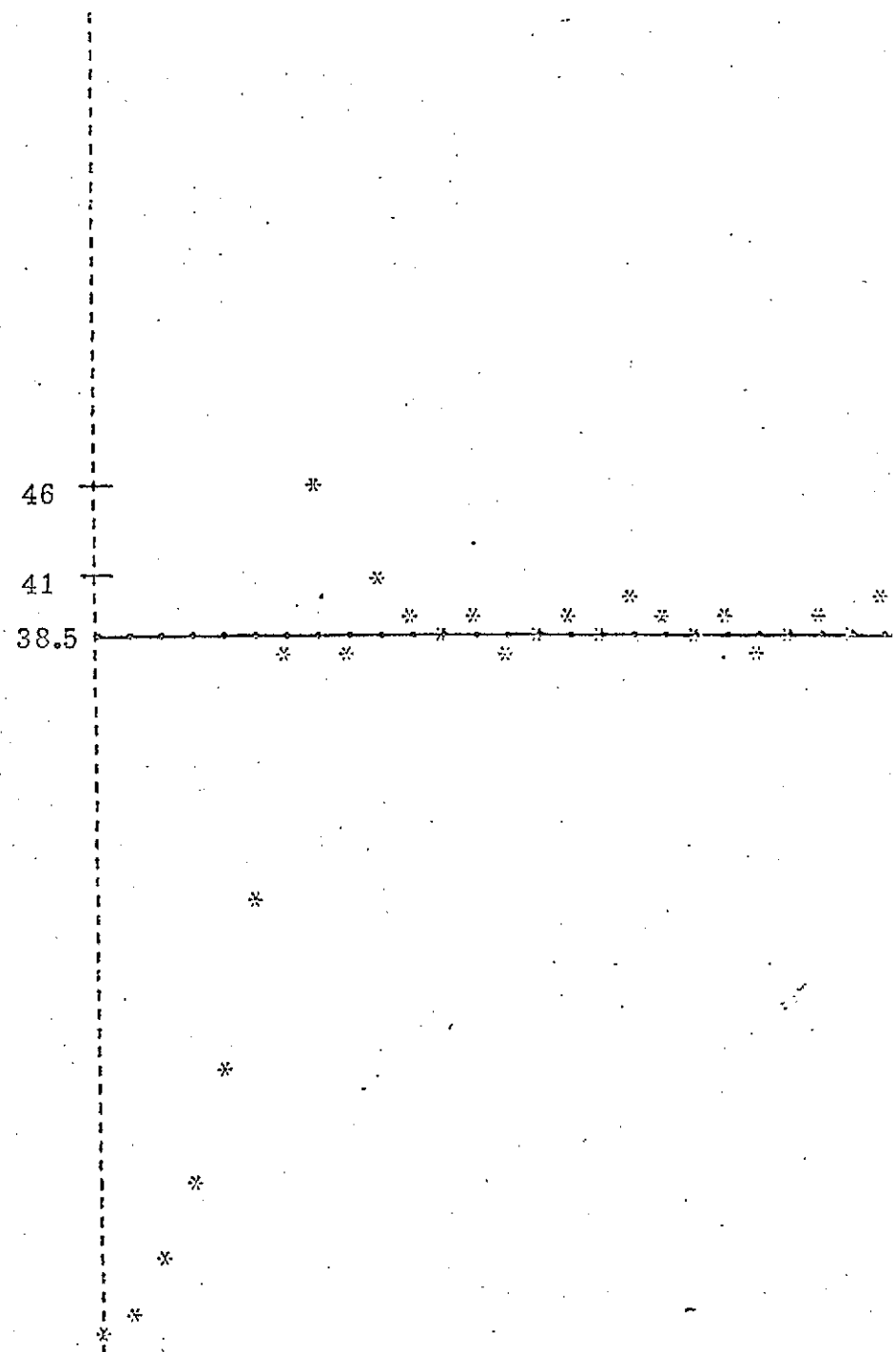


Fig. 6(a') The step response of a Song-Adaptive Delta Modulator with Overshoot Suppression.

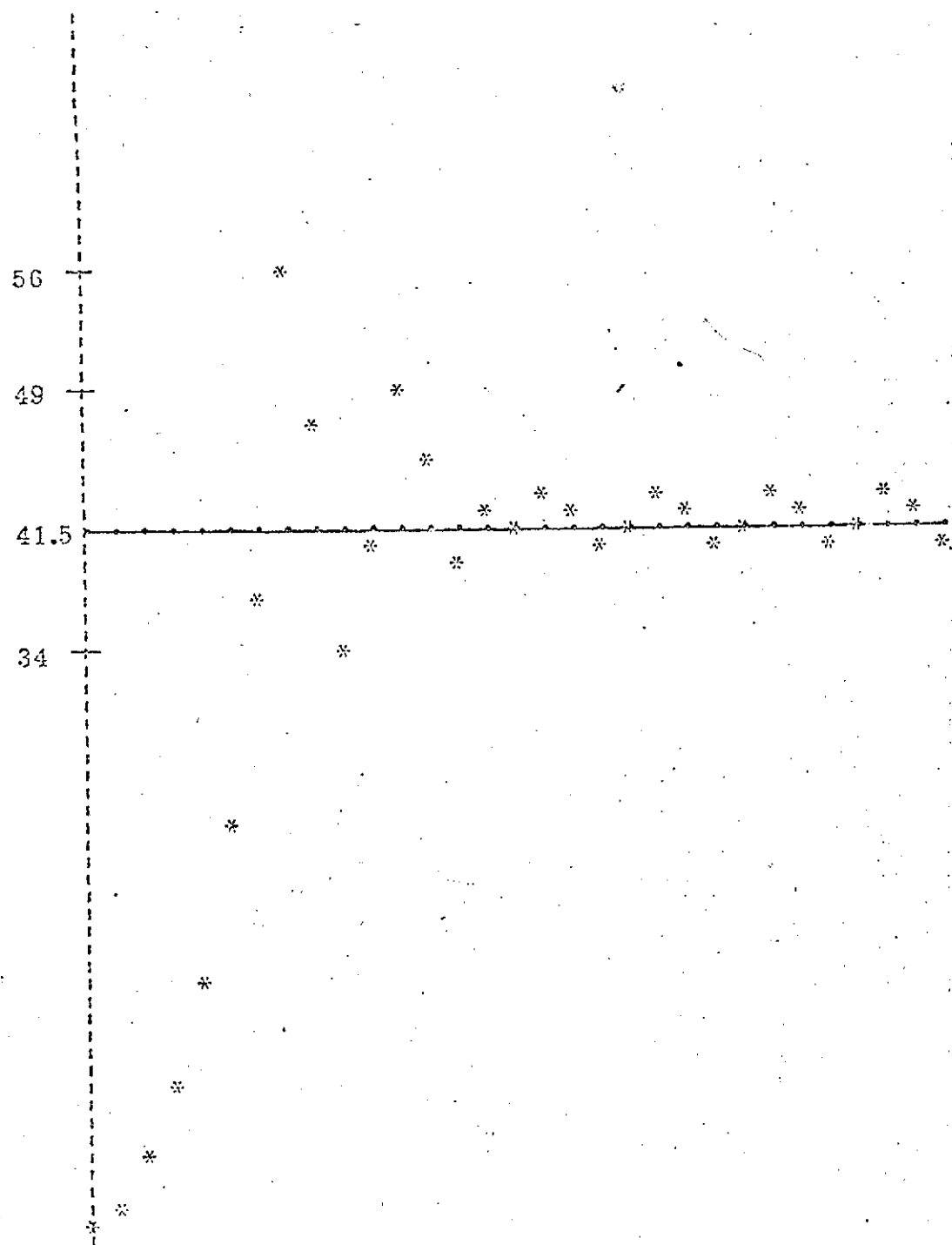


Fig. 6(b) The step response of a Song-Adaptive Delta Modulator without Overshoot Suppression.

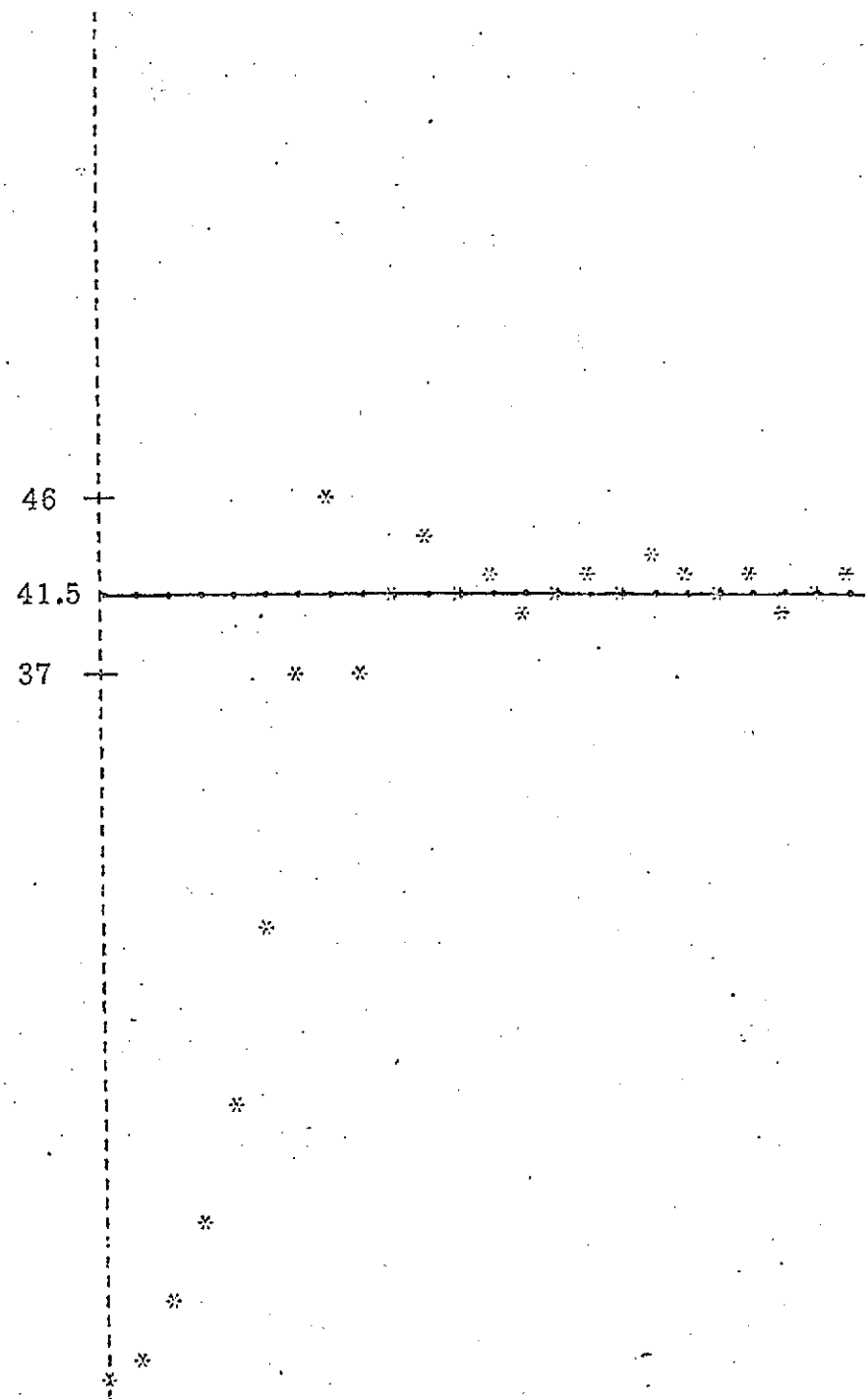


Fig. 6(b') The step response of a Song-Adaptive Delta Modulator with Overshoot Suppression.

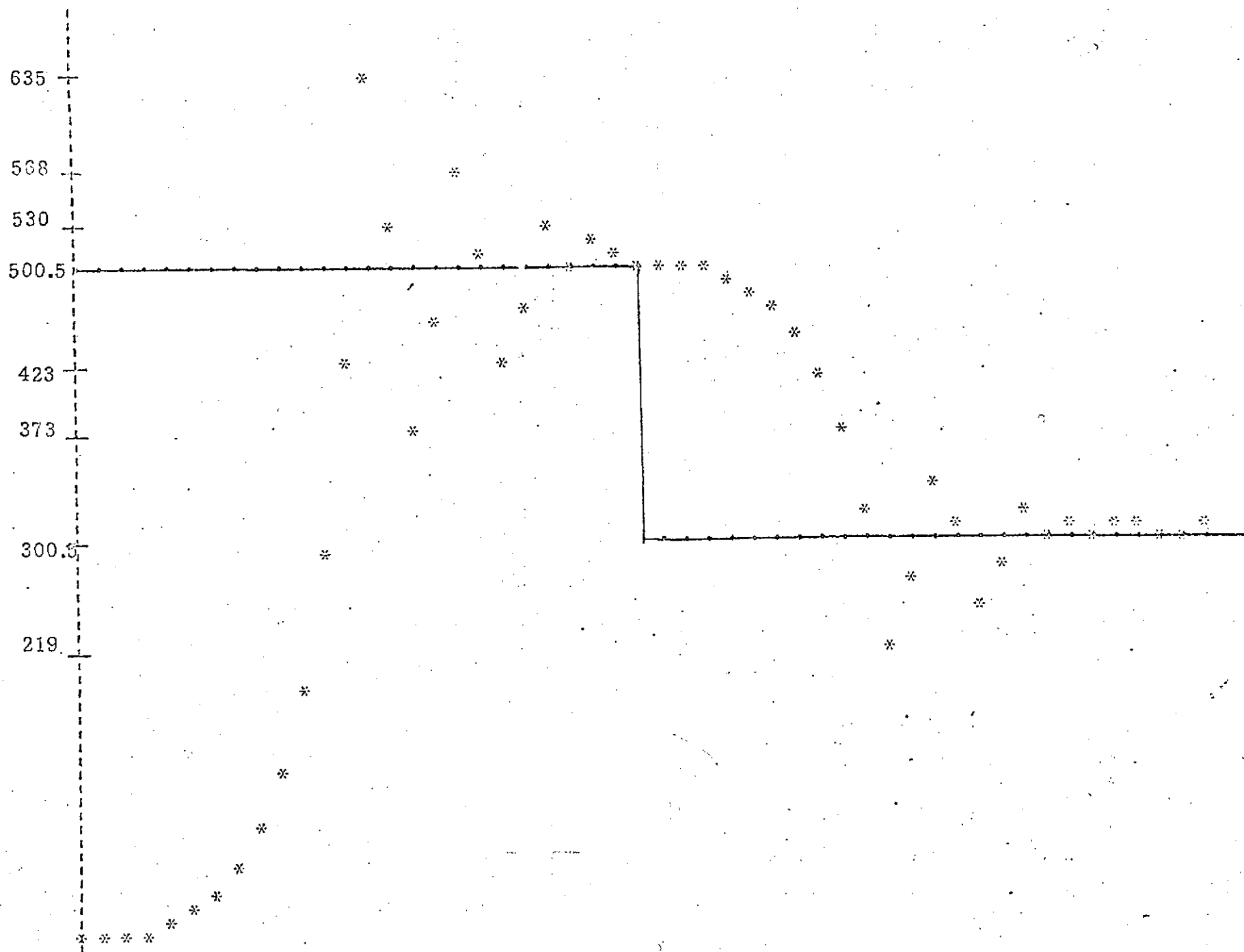
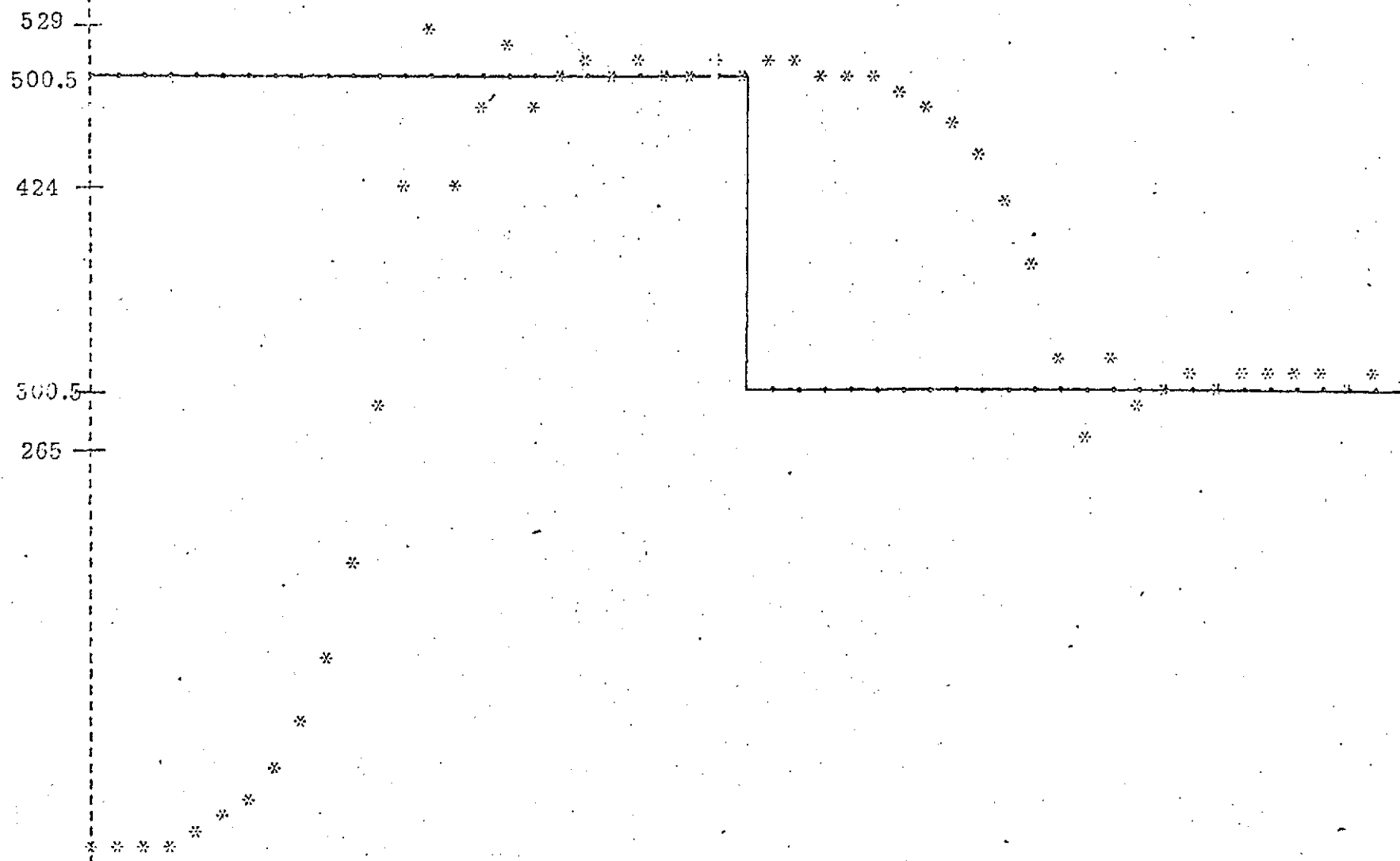


Fig. 7(a) The response of a Song-Adaptive Delta Modulator to abrupt level changes without Overshoot Suppression.



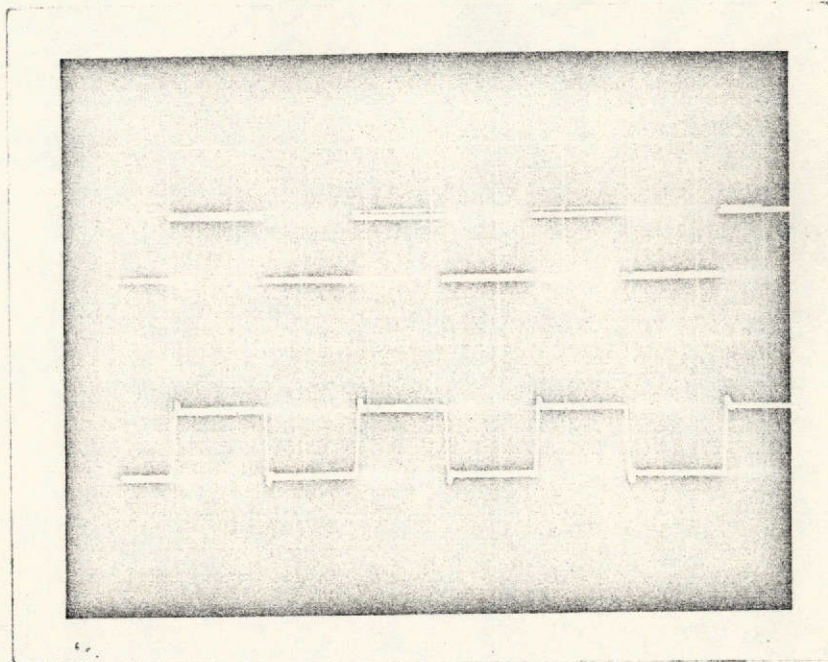


Figure 8(a) Upper Waveform: 350 Hz square wave ( $m(t)$  ).  
 Lower Waveform:  $m(t)$  bandlimited to 10 KHz.

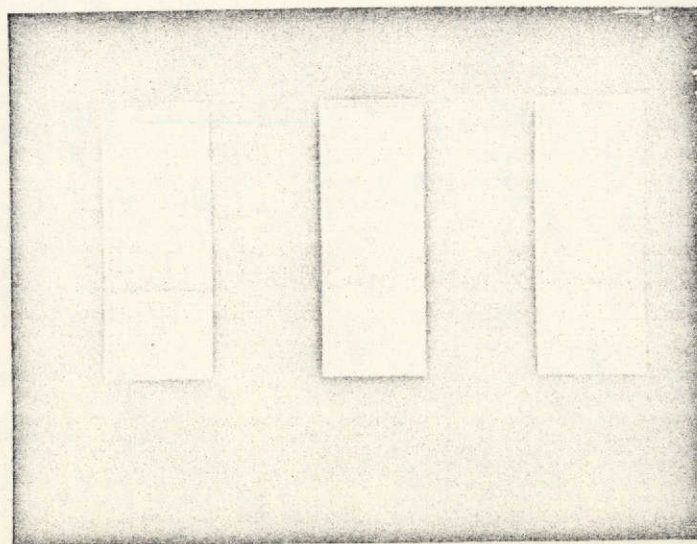


Figure 8(a') 100 line raster intensity modulated by bandlimited  $m(t)$ .



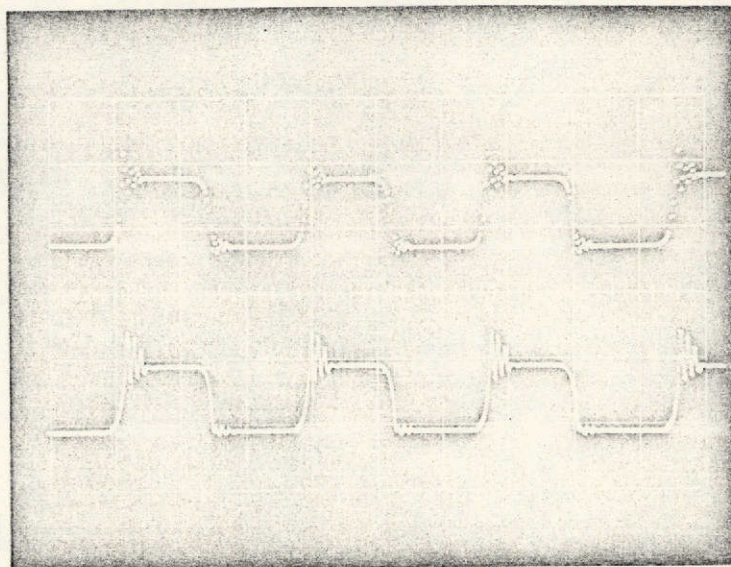


Figure 8(b) Upper Waveform: Bandlimited  $m(t)$  after DM processing (without OSS) at  $f_s = 40 \text{ KHz}$ .

Lower Waveform: Upper Waveform low-pass filtered ( $f_m = 10 \text{ KHz}$ ) =  $x(t)$ .

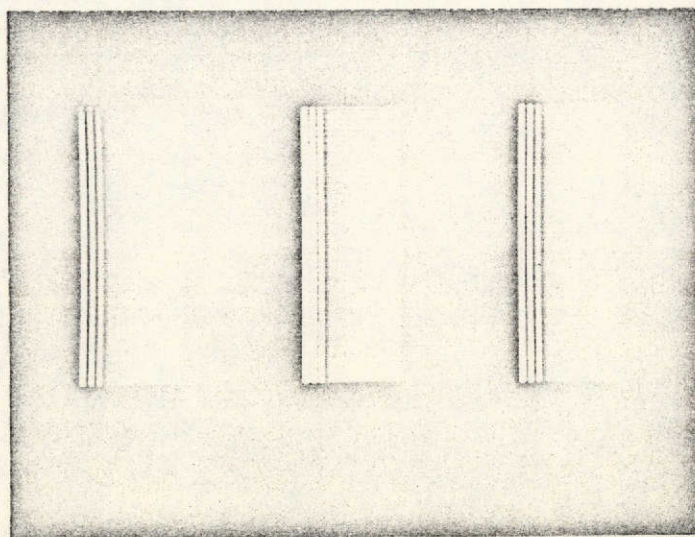


Figure 8(b') 100 line raster intensity modulated by  $x(t)$ .



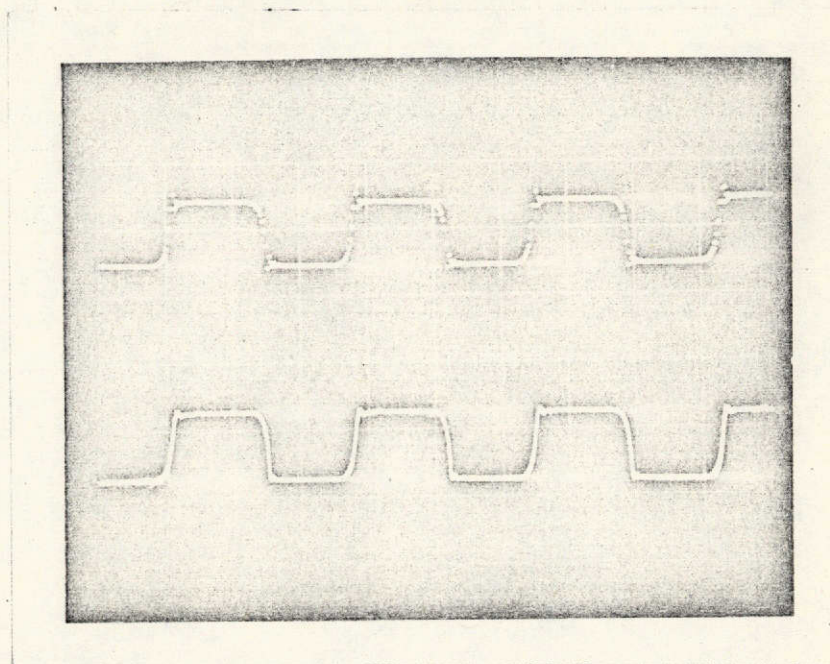


Figure 8(c) Upper Waveform: Bandlimited  $m(t)$  after DM processing (with OSS)  
 at  $f_s = 40\text{KHz}$ .  
 Lower Waveform: Upper Waveform low-pass filtered ( $f_m = 10\text{KHz}$ ) =  $X'(t)$ .

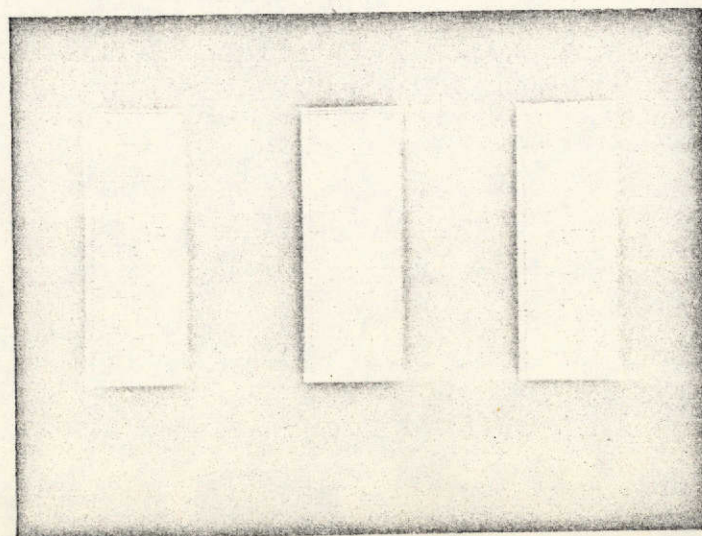


Figure 8(c') 100 line raster intensity modulated by  $X'(t)$ .





Figure 9(a) Original analog picture sampled at 400 samples/line and 170 lines/frame.



Figure 9(b) ADM processed original without OSS (64 grey levels).



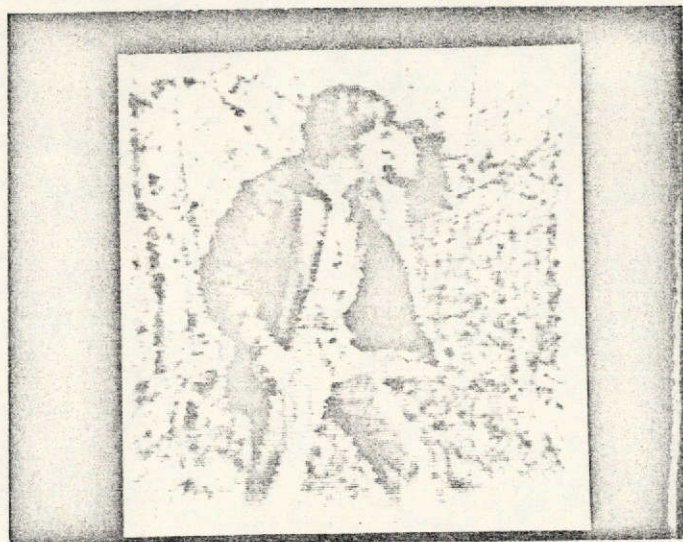


Figure 9(c) ADM processed original with OSS but without a threshold.



ORIGINAL PAGE IS  
OF POOR QUALITY

Figure 9(d) ADM processed original with OSS and with a threshold set at 8 grey levels.

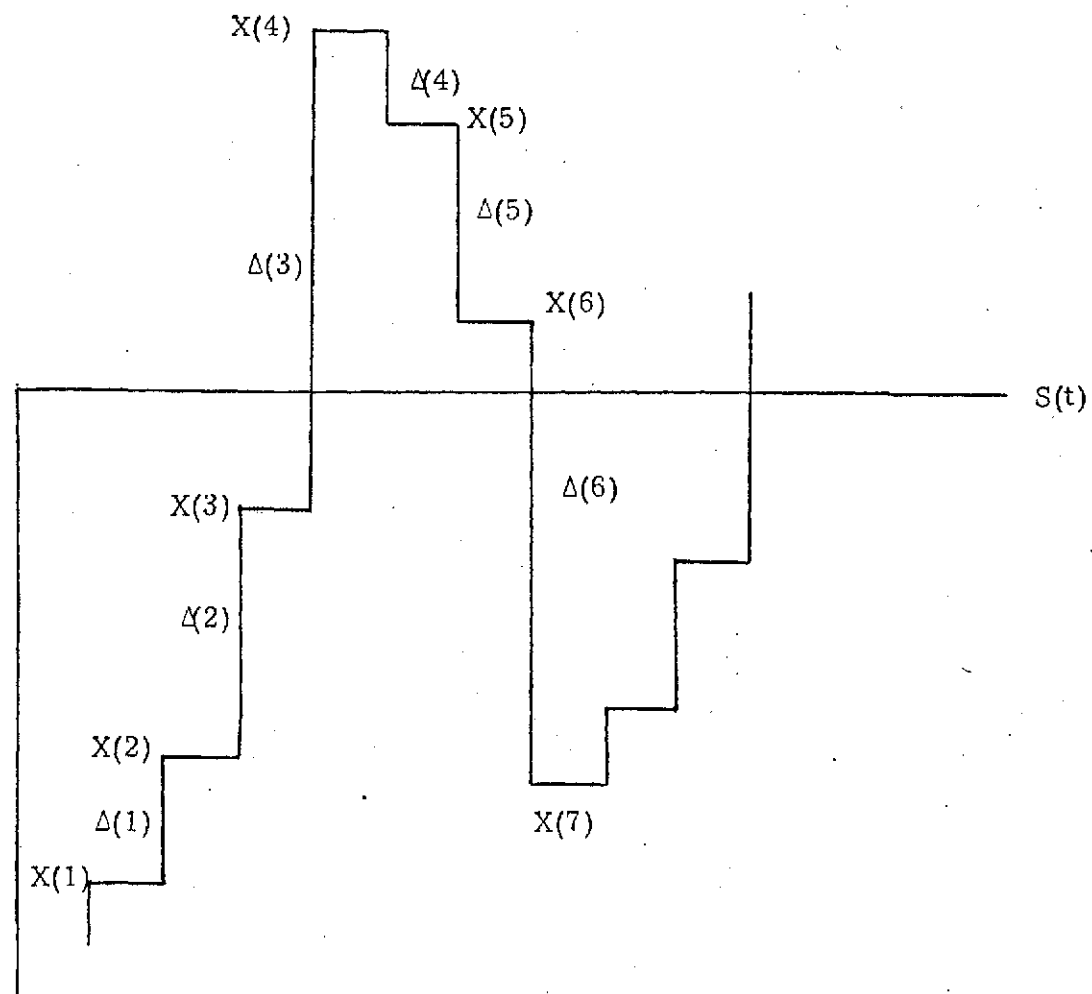


Figure 10 Stable Step Response with  $n = 3$ .

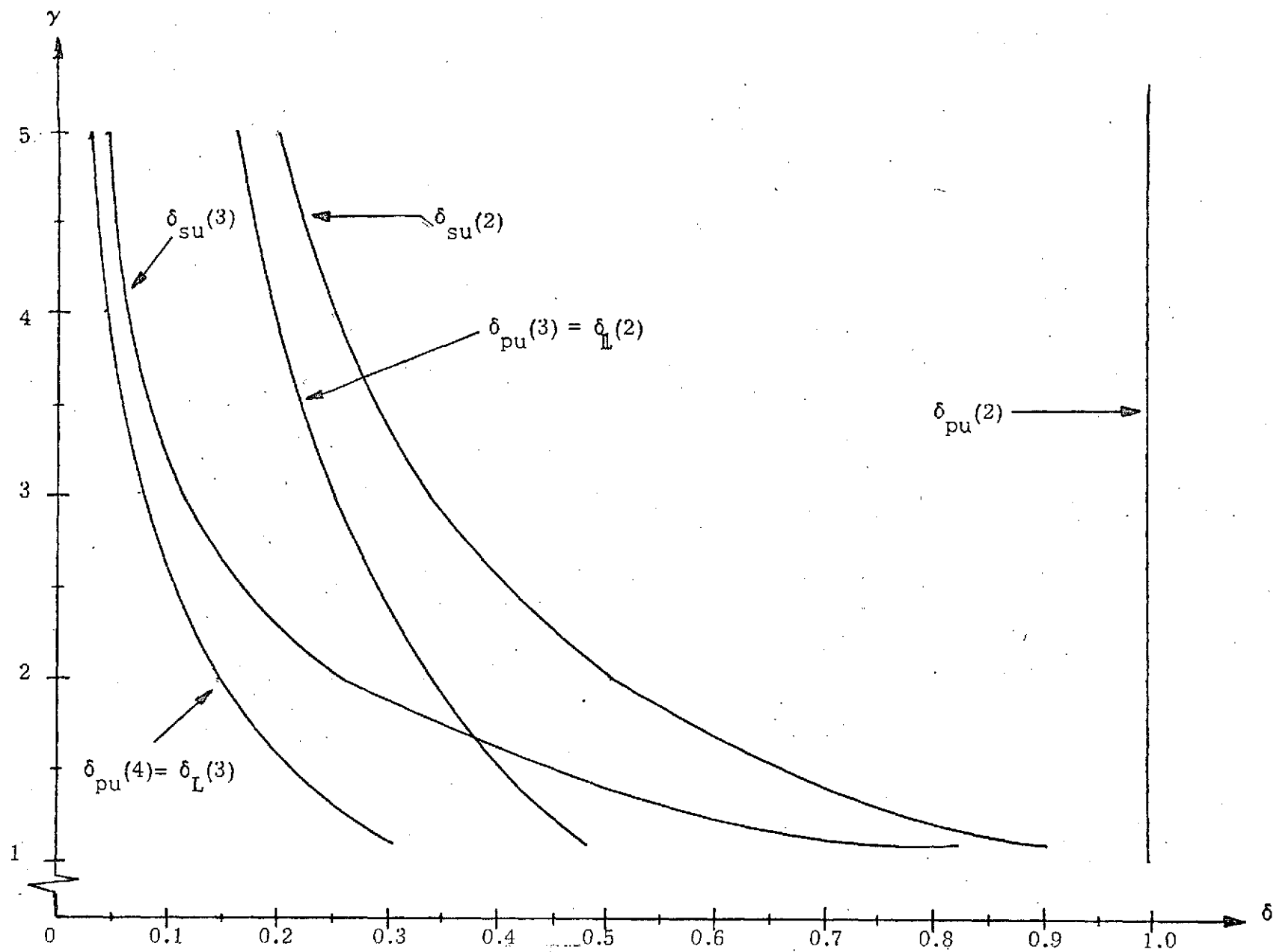


Figure 11 Stability Regions for  $n=2$ , and  $3$ . NOTE: (a)  $\delta_{pu}(n) = \frac{1-\gamma}{1-\gamma^{n-1}}$ , (b)  $\delta_{su}(n) = \gamma^{1-n}$ ,

(c)  $\delta_L(n) = \frac{1-\gamma}{1-\gamma^n}$

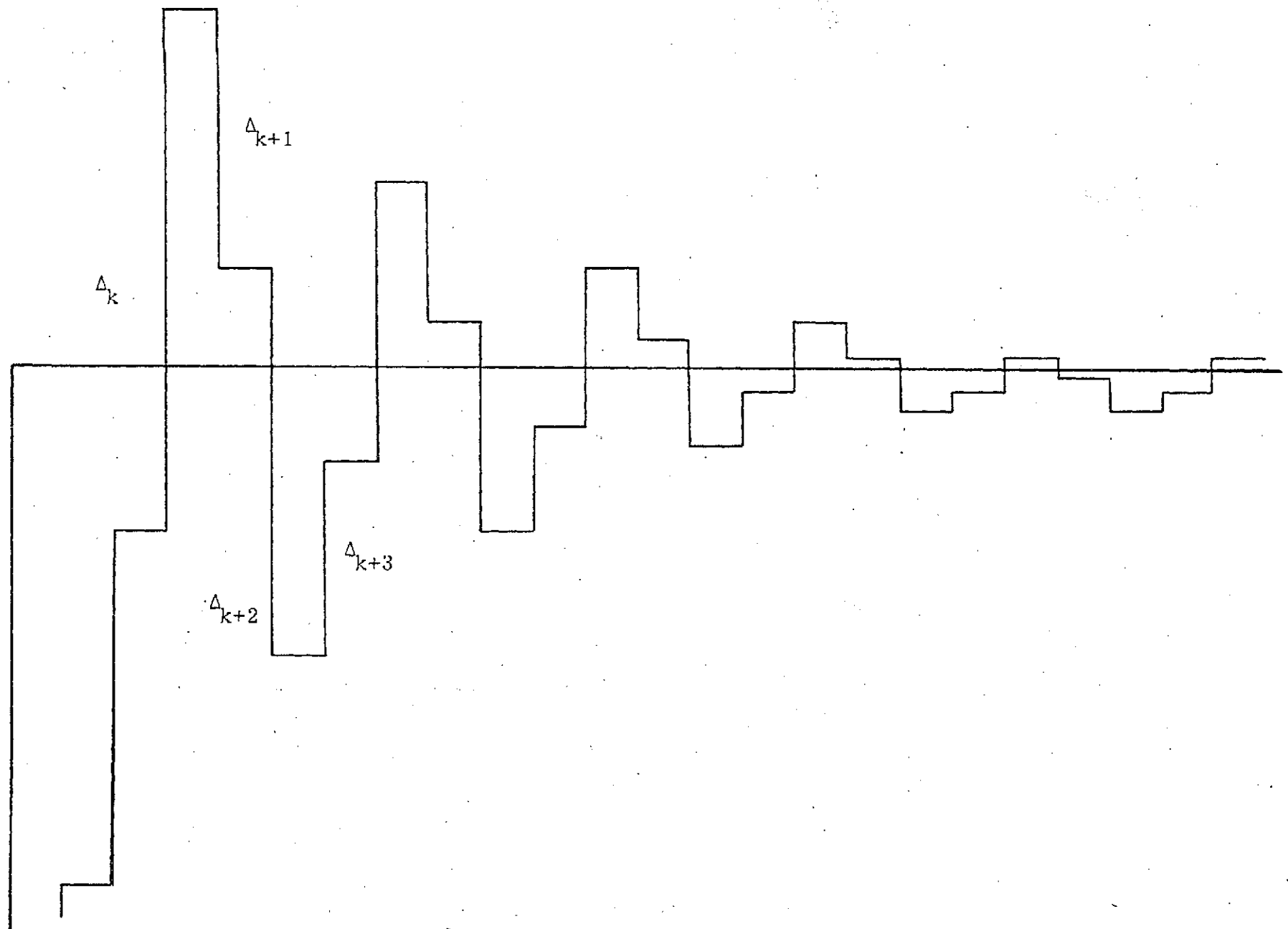


Figure 12(a) Step response without OSS used to upper bound settling time.

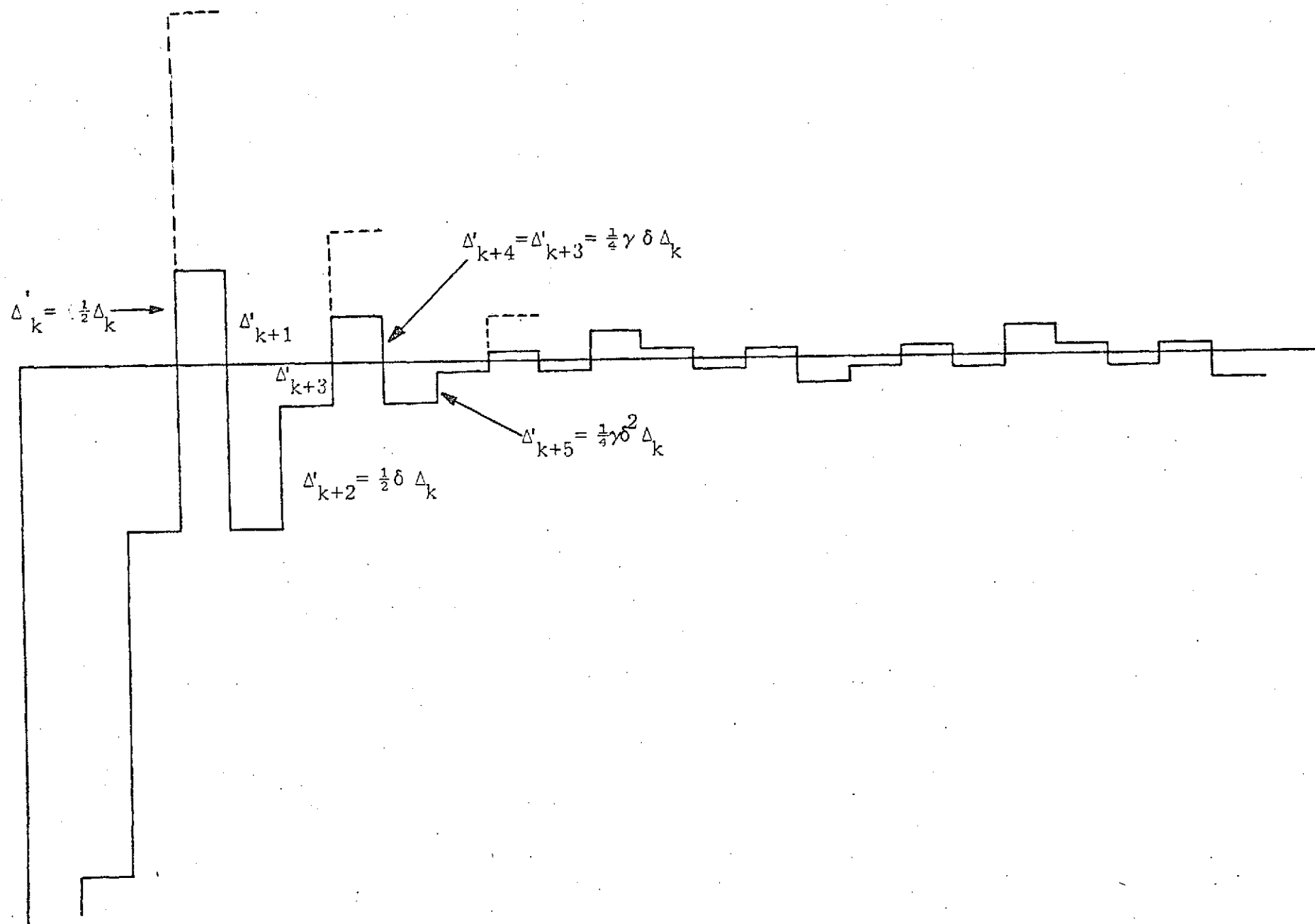


Figure 12(b) Step response with OSS used to upper bound settling time.

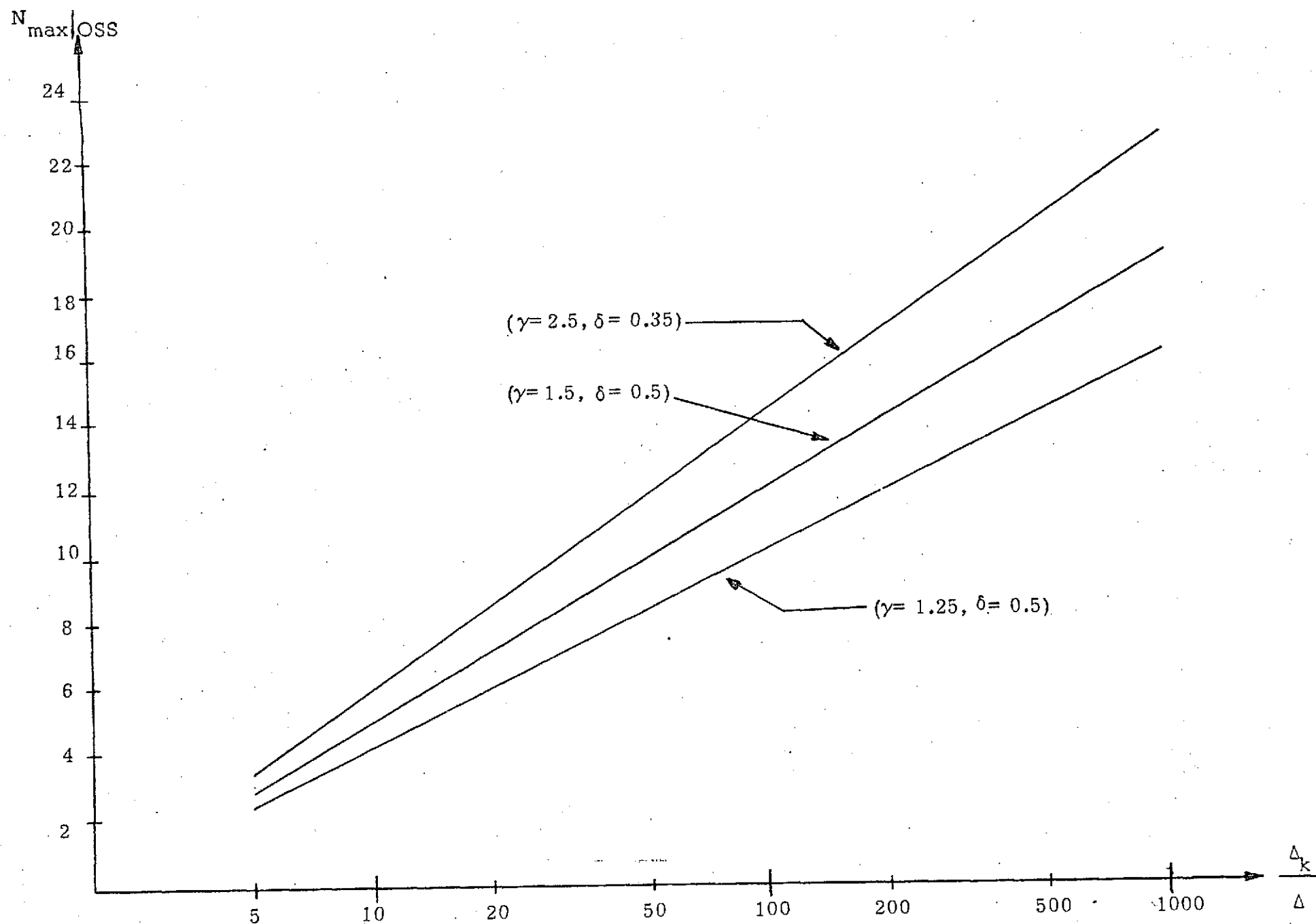


Figure 13 Settling time as a function of overshoot step size when OSS is implemented.



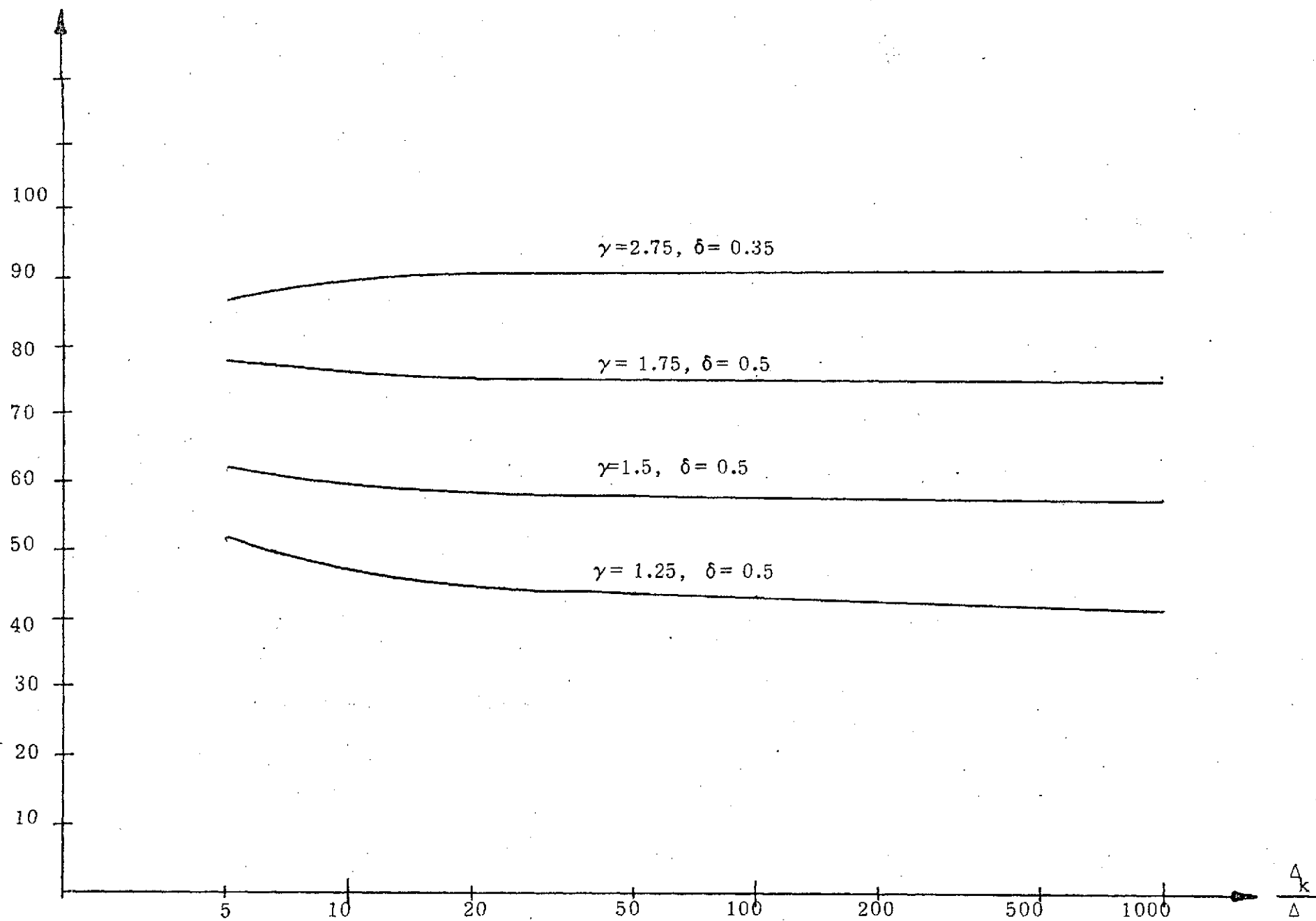


Figure 14 Percentage improvement in Settling Time due to OSS.

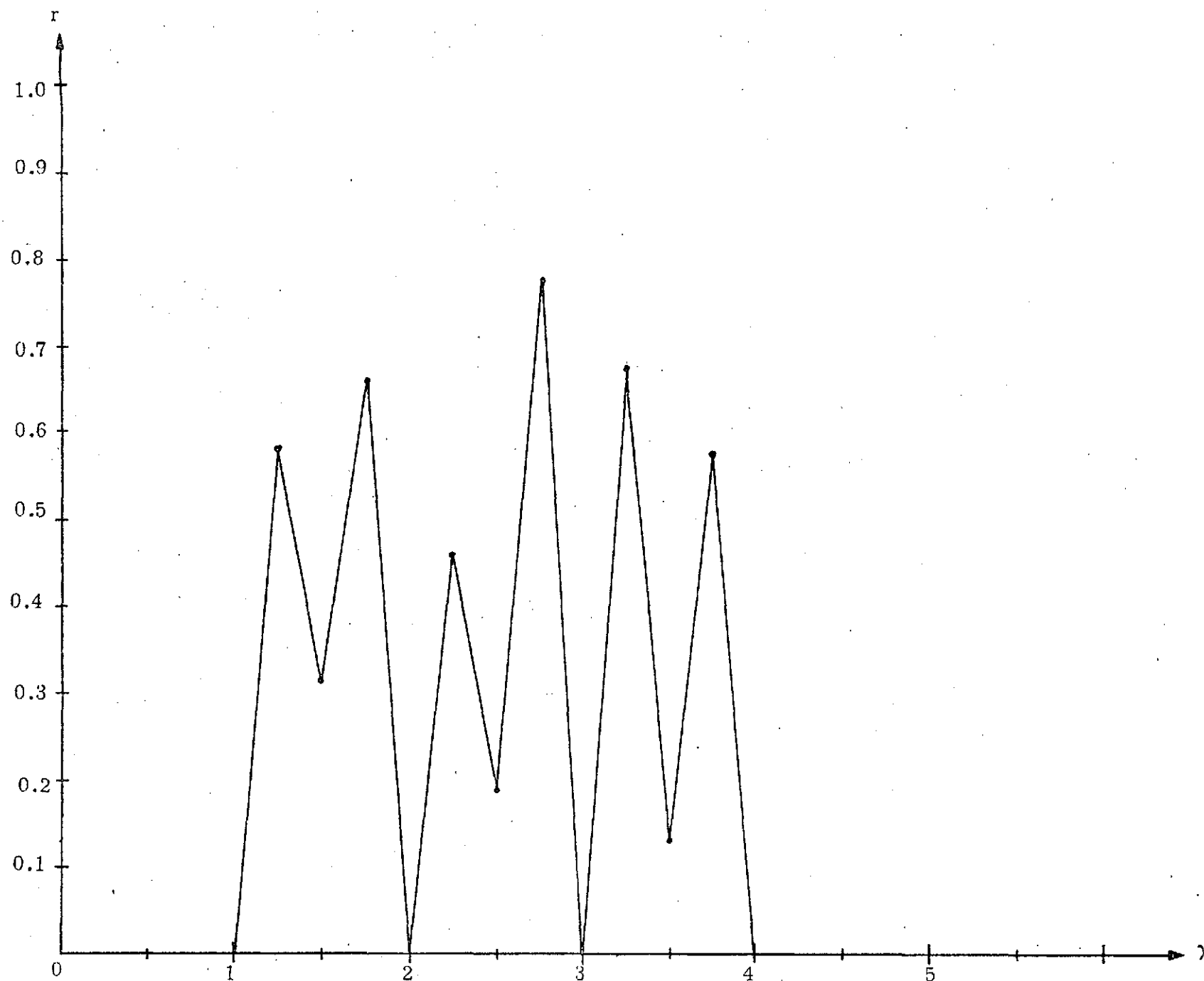


Figure 15 The local round-off error  $r$  versus the step size adaptation parameter  $\gamma$ , for a 10 bit digital implementation.

K= 0.000	SK= 500.5	XK= 0.000	EKE= 1.000
K= 1.000	SK= 500.5	XK= 2.000	EKE= 1.000
K= 2.000	SK= 500.5	XK= 5.000	EKE= 1.000
K= 3.000	SK= 500.5	XK= 9.000	EKE= 1.000
K= 4.000	SK= 500.5	XK= 15.00	EKE= 1.000
K= 5.000	SK= 500.5	XK= 24.00	EKE= 1.000
K= 6.000	SK= 500.5	XK= 37.00	EKE= 1.000
K= 7.000	SK= 500.5	XK= 56.00	EKE= 1.000
K= 8.000	SK= 500.5	XK= 84.00	EKE= 1.000
K= 9.000	SK= 500.5	XK= 126.0	EKE= 1.000
K= 10.00	SK= 500.5	XK= 189.0	EKE= 1.000
K= 11.00	SK= 500.5	XK= 283.0	EKE= -1.000
K= 12.00	SK= 500.5	XK= 424.0	EKE= -1.000
K= 13.00	SK= 500.5	XK= 635.0	EKE= -1.000
K= 14.00	SK= 500.5	XK= 936.0	EKE= -1.000
K= 15.00	SK= 500.5	XK= 1373.0	EKE= 1.000
K= 16.00	SK= 500.5	XK= 2051.0	EKE= 1.000
K= 17.00	SK= 500.5	XK= 3068.0	EKE= -1.000
K= 18.00	SK= 500.5	XK= 4510.0	EKE= -1.000
K= 19.00	SK= 500.5	XK= 6623.0	EKE= 1.000
K= 20.00	SK= 500.5	XK= 9666.0	EKE= 1.000
K= 21.00	SK= 500.5	XK= 14030.0	EKE= -1.000
K= 22.00	SK= 500.5	XK= 20380.0	EKE= 1.000
K= 23.00	SK= 500.5	XK= 29514.0	EKE= -1.000
K= 24.00	SK= 500.5	XK= 42660.0	EKE= -1.000
K= 25.00	SK= 500.5	XK= 62440.0	EKE= 1.000
K= 26.00	SK= 300.5	XK= 90300.0	EKE= -1.000
K= 27.00	SK= 300.5	XK= 130700.0	EKE= -1.000
K= 28.00	SK= 300.5	XK= 190300.0	EKE= -1.000
K= 29.00	SK= 300.5	XK= 274700.0	EKE= -1.000
K= 30.00	SK= 300.5	XK= 399800.0	EKE= -1.000
K= 31.00	SK= 300.5	XK= 582500.0	EKE= -1.000
K= 32.00	SK= 300.5	XK= 846600.0	EKE= -1.000
K= 33.00	SK= 300.5	XK= 1238000.0	EKE= -1.000
K= 34.00	SK= 300.5	XK= 1813000.0	EKE= -1.000
K= 35.00	SK= 300.5	XK= 2646000.0	EKE= -1.000
K= 36.00	SK= 300.5	XK= 3899000.0	EKE= 1.000
K= 37.00	SK= 300.5	XK= 5746000.0	EKE= 1.000
K= 38.00	SK= 300.5	XK= 8466000.0	EKE= -1.000
K= 39.00	SK= 300.5	XK= 12380000.0	EKE= -1.000
K= 40.00	SK= 300.5	XK= 18130000.0	EKE= 1.000
K= 41.00	SK= 300.5	XK= 26460000.0	EKE= 1.000
K= 42.00	SK= 300.5	XK= 38990000.0	EKE= -1.000
K= 43.00	SK= 300.5	XK= 57460000.0	EKE= -1.000
K= 44.00	SK= 300.5	XK= 84660000.0	EKE= -1.000
K= 45.00	SK= 300.5	XK= 123800000.0	EKE= 1.000
K= 46.00	SK= 300.5	XK= 181300000.0	EKE= -1.000
K= 47.00	SK= 300.5	XK= 264600000.0	EKE= -1.000
K= 48.00	SK= 300.5	XK= 389900000.0	EKE= 1.000
K= 49.00	SK= 300.5	XK= 574600000.0	EKE= 1.000
K= 50.00	SK= 300.5	XK= 846600000.0	EKE= -1.000

Table 1(a) Table of values for Fig. 7(a)

K= K'th sampling instant.

SK = input signal at time K.

XK = input estimate at time K.

EKE = sign bit at time K.

ORIGINAL PAGE IS  
OF POOR QUALITY

K= 0.000	SK= 500.5	XK= 0.000	EKU= 1.000	EKT= 1.000
K= 1.000	SK= 500.5	XK= 2.000	EKU= 1.000	EKT= 1.000
K= 2.000	SK= 500.5	XK= 5.000	EKU= 1.000	EKT= 1.000
K= 3.000	SK= 500.5	XK= 9.000	EKU= 1.000	EKT= 1.000
K= 4.000	SK= 500.5	XK= 15.00	EKU= 1.000	EKT= 1.000
K= 5.000	SK= 500.5	XK= 24.00	EKU= 1.000	EKT= 1.000
K= 6.000	SK= 500.5	XK= 37.00	EKU= 1.000	EKT= 1.000
K= 7.000	SK= 500.5	XK= 56.00	EKU= 1.000	EKT= 1.000
K= 8.000	SK= 500.5	XK= 84.00	EKU= 1.000	EKT= 1.000
K= 9.000	SK= 500.5	XK= 126.0	EKU= 1.000	EKT= 1.000
K= 10.00	SK= 500.5	XK= 189.0	EKU= 1.000	EKT= 1.000
K= 11.00	SK= 500.5	XK= 283.0	EKU= 1.000	EKT= 1.000
K= 12.00	SK= 500.5	XK= 424.0	EKU= 1.000	EKT= 1.000
K= 13.00	SK= 500.5	XK= 529.0	EKU= -1.000	EKT= -1.000
K= 14.00	SK= 500.5	XK= 424.0	EKU= 1.000	EKT= -1.000
K= 15.00	SK= 500.5	XK= 476.0	EKU= 1.000	EKT= 1.000
K= 16.00	SK= 500.5	XK= 515.0	EKU= -1.000	EKT= -1.000
K= 17.00	SK= 500.5	XK= 476.0	EKU= 1.000	EKT= -1.000
K= 18.00	SK= 500.5	XK= 495.0	EKU= 1.000	EKT= 1.000
K= 19.00	SK= 500.5	XK= 509.0	EKU= -1.000	EKT= -1.000
K= 20.00	SK= 500.5	XK= 495.0	EKU= 1.000	EKT= -1.000
K= 21.00	SK= 500.5	XK= 502.0	EKU= -1.000	EKT= -1.000
K= 22.00	SK= 500.5	XK= 499.0	EKU= 1.000	EKT= 1.000
K= 23.00	SK= 500.5	XK= 500.0	EKU= 1.000	EKT= 1.000
K= 24.00	SK= 500.5	XK= 501.0	EKU= -1.000	EKT= -1.000
K= 25.00	SK= 500.5	XK= 500.0	EKU= 1.000	EKT= -1.000
K= 26.00	SK= 300.5	XK= 502.0	EKU= -1.000	EKT= -1.000
K= 27.00	SK= 300.5	XK= 501.0	EKU= -1.000	EKT= -1.000
K= 28.00	SK= 300.5	XK= 499.0	EKU= -1.000	EKT= -1.000
K= 29.00	SK= 300.5	XK= 490.0	EKU= -1.000	EKT= -1.000
K= 30.00	SK= 300.5	XK= 492.0	EKU= -1.000	EKT= -1.000
K= 31.00	SK= 300.5	XK= 486.0	EKU= -1.000	EKT= -1.000
K= 32.00	SK= 300.5	XK= 477.0	EKU= -1.000	EKT= -1.000
K= 33.00	SK= 300.5	XK= 464.0	EKU= -1.000	EKT= -1.000
K= 34.00	SK= 300.5	XK= 445.0	EKU= -1.000	EKT= -1.000
K= 35.00	SK= 300.5	XK= 417.0	EKU= -1.000	EKT= -1.000
K= 36.00	SK= 300.5	XK= 375.0	EKU= -1.000	EKT= -1.000
K= 37.00	SK= 300.5	XK= 312.0	EKU= -1.000	EKT= -1.000
K= 38.00	SK= 300.5	XK= 265.0	EKU= 1.000	EKT= 1.000
K= 39.00	SK= 300.5	XK= 312.0	EKU= -1.000	EKT= 1.000
K= 40.00	SK= 300.5	XK= 280.0	EKU= 1.000	EKT= 1.000
K= 41.00	SK= 300.5	XK= 300.0	EKU= 1.000	EKT= 1.000
K= 42.00	SK= 300.5	XK= 308.0	EKU= -1.000	EKT= -1.000
K= 43.00	SK= 300.5	XK= 300.0	EKU= 1.000	EKT= -1.000
K= 44.00	SK= 300.5	XK= 304.0	EKU= -1.000	EKT= -1.000
K= 45.00	SK= 300.5	XK= 302.0	EKU= -1.000	EKT= -1.000
K= 46.00	SK= 300.5	XK= 301.0	EKU= 1.000	EKT= 1.000
K= 47.00	SK= 300.5	XK= 302.0	EKU= -1.000	EKT= 1.000
K= 48.00	SK= 300.5	XK= 300.0	EKU= 1.000	EKT= 1.000
K= 49.00	SK= 300.5	XK= 301.0	EKU= -1.000	EKT= -1.000

Table 1 (b) Table of values for Fig. 7(b)

EKU = sign bit used inside encoder and decoder.

EKT = sign bit transmitted from encoder to decoder.

ORIGINAL PAGE IS  
OF POOR QUALITY

I= 1	XRONI= 1	XNRONI= 1.0000	ECUMI= 0.0000
I= 2	XRONI= 3	XNRONI= 3.0000	ECUMI= 0.0000
I= 3	XRONI= 6	XNRONI= 6.0000	ECUMI= 0.0000
I= 4	XRONI= 10	XNRONI= 10.5000	ECUMI= 0.5000
I= 5	XRONI= 16	XNRONI= 17.2500	ECUMI= 1.2500
I= 6	XRONI= 25	XNRONI= 27.3750	ECUMI= 2.3750
I= 7	XRONI= 38	XNRONI= 42.5625	ECUMI= 4.5625
I= 8	XRONI= 57	XNRONI= 65.3438	ECUMI= 8.3438
I= 9	XRONI= 85	XNRONI= 99.5156	ECUMI= 14.5156
I= 10	XRONI= 127	XNRONI= 150.7740	ECUMI= 23.7734
I= 11	XRONI= 190	XNRONI= 227.6600	ECUMI= 37.6602
I= 12	XRONI= 284	XNRONI= 342.9900	ECUMI= 58.9902
I= 13	XRONI= 425	XNRONI= 515.9850	ECUMI= 90.9854
I= 14	XRONI= 636	XNRONI= 775.4780	ECUMI= 139.4780
I= 15	XRONI= 952	XNRONI= 1164.7200	ECUMI= 212.7170
I= 16	XRONI= 1426	XNRONI= 1748.5800	ECUMI= 322.5760
I= 17	XRONI= 2137	XNRONI= 2624.3600	ECUMI= 487.3630
I= 18	XRONI= 3203	XNRONI= 3938.0500	ECUMI= 735.0440
I= 19	XRONI= 4802	XNRONI= 5908.5700	ECUMI= 1106.5700
I= 20	XRONI= 7200	XNRONI= 8864.3500	ECUMI= 1664.3500

\*

Table 2 Sampling Time vs. Level, with and without truncation, and Sampling Time vs. Cumulative Error ( $\gamma = \alpha + \beta = 1.5$ ).

Note: I = Sampling Instant

XRONI = Level at time I with truncation

XNRONI = Level at time I without truncation.

ECUMI = Cumulative error at time I

ORIGINAL PAGE IS  
OF POOR QUALITY

Table 3

$r$  versus  $\gamma$  as given by Eq (38) for a 10 bit digital implementation.

$\gamma$	$r$	$\Delta$	$M$
1.00	0	1	-
1.25	0.5765	1	4
1.50	0.3122	1	2
1.75	0.6604	1	2
2.00	0	1	2
2.25	0.4581	1	2
2.50	0.1831	1	2
2.75	0.7689	1	2
3.00	0	1	2
3.25	0.6683	1	2
3.50	0.1311	1	2
3.75	0.5733	1	2
4.00	0	1	2

THIS PROGRAM TABULATES LEVEL VS. RISETIME WITH GAMMA AND R  
AS PARAMETERS.

LEVEL=	1.00	I1=	1.36	I2=	0.86	I3=	0.51	I4=	0.57
LEVEL=	2.00	I1=	1.95	I2=	1.38	I3=	1.71	I4=	2.08
LEVEL=	3.00	I1=	2.42	I2=	1.84	I3=	2.11	I4=	2.02
LEVEL=	4.00	I1=	2.82	I2=	2.24	I3=	2.46	I4=	2.40
LEVEL=	5.00	I1=	3.17	I2=	2.60	I3=	2.78	I4=	2.74
LEVEL=	6.00	I1=	3.47	I2=	2.92	I3=	3.07	I4=	3.04
LEVEL=	7.00	I1=	3.74	I2=	3.21	I3=	3.33	I4=	3.31
LEVEL=	8.00	I1=	3.98	I2=	3.47	I3=	3.58	I4=	3.56
LEVEL=	9.00	I1=	4.20	I2=	3.71	I3=	3.80	I4=	3.79
LEVEL=	10.00	I1=	4.40	I2=	3.93	I3=	4.01	I4=	4.00
LEVEL=	11.00	I1=	4.59	I2=	4.14	I3=	4.21	I4=	4.20
LEVEL=	12.00	I1=	4.76	I2=	4.33	I3=	4.39	I4=	4.38
LEVEL=	13.00	I1=	4.92	I2=	4.50	I3=	4.56	I4=	4.55
LEVEL=	14.00	I1=	5.08	I2=	4.67	I3=	4.72	I4=	4.71
LEVEL=	15.00	I1=	5.22	I2=	4.83	I3=	4.87	I4=	4.87
LEVEL=	16.00	I1=	5.35	I2=	4.97	I3=	5.01	I4=	5.01
LEVEL=	17.00	I1=	5.48	I2=	5.11	I3=	5.15	I4=	5.15
LEVEL=	18.00	I1=	5.61	I2=	5.25	I3=	5.28	I4=	5.28
LEVEL=	19.00	I1=	5.72	I2=	5.37	I3=	5.40	I4=	5.40
LEVEL=	20.00	I1=	5.83	I2=	5.49	I3=	5.52	I4=	5.52
LEVEL=	21.00	I1=	5.94	I2=	5.61	I3=	5.64	I4=	5.63
LEVEL=	22.00	I1=	6.04	I2=	5.72	I3=	5.74	I4=	5.74
LEVEL=	23.00	I1=	6.14	I2=	5.83	I3=	5.85	I4=	5.85
LEVEL=	24.00	I1=	6.23	I2=	5.93	I3=	5.95	I4=	5.95
LEVEL=	25.00	I1=	6.32	I2=	6.03	I3=	6.04	I4=	6.04
LEVEL=	26.00	I1=	6.41	I2=	6.12	I3=	6.14	I4=	6.14
LEVEL=	27.00	I1=	6.50	I2=	6.21	I3=	6.23	I4=	6.23
LEVEL=	28.00	I1=	6.58	I2=	6.30	I3=	6.31	I4=	6.31
LEVEL=	29.00	I1=	6.66	I2=	6.38	I3=	6.40	I4=	6.40
LEVEL=	30.00	I1=	6.73	I2=	6.47	I3=	6.48	I4=	6.48
LEVEL=	31.00	I1=	6.81	I2=	6.54	I3=	6.56	I4=	6.56
LEVEL=	32.00	I1=	6.88	I2=	6.62	I3=	6.64	I4=	6.63
LEVEL=	33.00	I1=	6.95	I2=	6.70	I3=	6.71	I4=	6.71
LEVEL=	34.00	I1=	7.02	I2=	6.77	I3=	6.78	I4=	6.78
LEVEL=	35.00	I1=	7.08	I2=	6.84	I3=	6.85	I4=	6.85
LEVEL=	36.00	I1=	7.15	I2=	6.91	I3=	6.92	I4=	6.92
LEVEL=	37.00	I1=	7.21	I2=	6.98	I3=	6.99	I4=	6.99
LEVEL=	38.00	I1=	7.27	I2=	7.04	I3=	7.05	I4=	7.05
LEVEL=	39.00	I1=	7.33	I2=	7.11	I3=	7.12	I4=	7.12
LEVEL=	40.00	I1=	7.39	I2=	7.17	I3=	7.18	I4=	7.18
LEVEL=	41.00	I1=	7.45	I2=	7.23	I3=	7.24	I4=	7.24
LEVEL=	42.00	I1=	7.50	I2=	7.29	I3=	7.30	I4=	7.30
LEVEL=	43.00	I1=	7.56	I2=	7.35	I3=	7.35	I4=	7.35
LEVEL=	44.00	I1=	7.61	I2=	7.40	I3=	7.41	I4=	7.41
LEVEL=	45.00	I1=	7.66	I2=	7.46	I3=	7.47	I4=	7.47
LEVEL=	46.00	I1=	7.72	I2=	7.51	I3=	7.52	I4=	7.52
LEVEL=	47.00	I1=	7.77	I2=	7.57	I3=	7.57	I4=	7.57
LEVEL=	48.00	I1=	7.81	I2=	7.62	I3=	7.62	I4=	7.62
LEVEL=	49.00	I1=	7.86	I2=	7.67	I3=	7.67	I4=	7.67
LEVEL=	50.00	I1=	7.91	I2=	7.72	I3=	7.72	I4=	7.72

Table 4 LEVEL vs. RISETIME given by Eq.(39) for the first 4 iterations.

Note that for LEVEL > 5 convergence occurs by the second iteration, I2.

## II. The Effect of Channel Errors on DM Encoded Video Signals

### ABSTRACT

Although Adaptive Delta Modulation (ADM) systems have found wide application in source encoding of voice, problems arise when they are used as source encoders for video signals. The difficulty in using an ADM in video applications results from the fact that transmission errors in ADM channels cause permanent d.c. shifts in the received video signal. These level shifts inflict serious damage to the quality of the received picture by producing bright (or dark) horizontal streaks across the picture.

A two step technique is proposed for overcoming the above difficulty. The first step requires the encoder to periodically send information about the signal level to the receiver via a PCM word. The second step involves a line to line correlation technique which detects error streaks and replaces the streak by the average intensity of the adjacent lines.

### INTRODUCTION - THE SONG ADM

The Song ADM shown in Fig 1 was used to encode the pictures presented in this paper. The following is an explanation of the Song ADM algorithm. The analog video signal,  $S(t)$ , is sampled at the  $k$ th instant of time, and converted into a digital signal  $S_k$ . This signal is then compared to  $X_k$ . The output,  $e_k$ , is obtained from the relation.

$$e_k = \text{sgn} (S_k - X_k) \quad (1)$$

and is transmitted over the channel.  $X_k$ , referred to as the transmitter estimate, is obtained using the recursive equation



$$X_k = X_{k-1} + \Delta_k \quad (2)$$

and  $\Delta_k$ , referred to as the current step size, is generated by

$$\Delta_k = \begin{cases} |\Delta_{k-1}| (e_{k-1} + \frac{1}{2} e_{k-2}) & ; \quad |\Delta_{k-1}| \geq 2\Delta \\ 2\Delta e_{k-1} & ; \quad |\Delta_{k-1}| < 2\Delta \end{cases} \quad (3)$$

where  $\Delta$  is a constant and is the minimum step-size of the system.

The decoder for the Song ADM is just the feedback loop of the encoder. the decoder reconstructs the estimate,  $X_k$ , from the  $e_k$  pulse train and converts  $X_k$  to an analog signal  $\hat{S}(t)$ . The quality of the received picture as compared to that of the transmitted picture depends upon how closely  $\hat{S}(t)$  approximates  $S(t)$  the original signal.

The effectiveness of the Song ADM as a video source encoder can be judged by comparing the picture in Fig 2, encoder by PCM, with the picture in Fig 3 encoded by the Song ADM. In Fig 3, a, b, and c the sampling rate of the ADM is 5, 4 and 3 times the nyquist rate of the video signal, respectively. Since the Song ADM transmits 1 bit per sample the number of bits per pixel in Fig 3a, b and c is also 5, 4 and 3. At these sampling rates the Song ADM produces pictures similar in quality to PCM.

If noise is introduced into the channel, the ADM encoded picture severely degrades, as shown in Fig 6b. This same noise rate would have little effect on the PCM channel. If the Song ADM is to be used as a video encoder over a noisy channel an error correcting scheme must be included to reduce the effects of channel errors on the received picture. The rest of this paper concerns itself with analyzing the effects of channel errors on the Song ADM and presents an error correcting scheme which minimizes the effects of channel errors.

### EFFECTS OF CHANNEL ERRORS ON THE SONG ADM

The response of the Song ADM to a voltage step input in the presence of channel errors is shown in Fig 4. The solid lines represent the signal,  $x_k$ , at the decoder in the absence of channel errors, while the dashed lines represent the signal,  $x_k$ , corrupted by a single channel error. It is obvious from Fig 4 that channel errors cause a permanent and significant DC shift in the received signal. We also observe that 50% of the time channel errors cause errors in the step size,  $\Delta_k$ , of the ADM (see Eq 3 and Fig 4).

We will now show that step size errors in an ADM become self correcting if the ADM tracks a constant DC level and if the constant DC level persists long enough. Observe from Fig 4 and Eqs 1, 2 and 3 that the step size,  $\Delta_k$ , will decrease while the ADM is tracking a constant DC level. If the constant signal level persists long enough the ADM reaches the minimum step size,  $\Delta$ , for both the corrected signal and the error free signal. When this condition occurs, the step size error has corrected itself since both the corrupted signal and error free signal will have the same step size namely  $\Delta$ , the minimum step size.

The number of transmitted bits,  $(e_k)$ , that will reach the receiver after the occurrence of a channel error, but before the step size corrects itself, has an upper bound for a Song ADM tracking a constant DC level. This upper bound can be shown to be

$$\begin{aligned}
 (.75) \quad \frac{N}{2} &= \Delta / \Delta_k \\
 \text{or} \quad N &= \frac{2 \ln \Delta / \Delta_k}{\ln .75}
 \end{aligned}
 \tag{4}$$

where  $N$  = the number of transmitted bits until step size correction occurs.

For the pictures shown in Fig 6 the worst case parameters for Eq 4, are  $\Delta = 1$  and  $\Delta_{\max} = 31$  and yield an  $N$  equal to 24.

Since there are five bits per pixel in the pictures of Fig 6, any step size error which might occur will be corrected whenever 5 pixels in a row have the same value. Of course, most step size errors will not occur under the worst case conditions and will be corrected sooner. Several pixels of the same value in a row are a common occurrence in a picture; thus step size errors are undetectable in Fig 6.

### ERROR CORRECTING ALGORITHM

The preceeding discussion has shown that channel errors effect delta modulated encoded video signals by altering the DC level of the demodulated signal. A two part correction algorithm is proposed to minimize the effects of the channel errors on the video signal.

#### PART I

The first step in the correction process shown in Fig 5 reduces the light and dark bands of Fig 6b to short streaks as in Fig 6c. This is accomplished by sending to the receiver the value of the transmitter's current estimate,  $x_k$ , (as a PCM codeword) after every  $i$  delta modulator bits have been transmitted. Upon receiving  $x_k$ , the receiver modifies its own current estimate to agree with  $x_k$ . Once the receiver's current estimate agrees with the transmitter's current estimate, all DC shifts caused by past channel errors are eliminated from future estimates. However, DC shifts caused by past channel errors are not eliminated from past receiver estimates. Thus, a single channel error may still cause a DC shift in at most  $i$  consecutive receiver estimates with this type of correction algorithm.

To implement this correction algorithm the channel bit rate must be increased to accommodate the periodic sending of the transmitters estimate. If the bit rate of the channel without correction is  $f_s$ , the sampling rate of the delta modulator, then the new channel bit rate  $f'_s$  is given by

$$f'_s = f_s \left(1 + \frac{b}{i}\right) \quad (5)$$

where 'b' is equal to the number of bits of the transmitters estimate that is sent to the receiver as a PCM word.

From Fig 5 it is apparent that  $i$  should be as large as possible and  $b$  as small as possible to minimize  $f'_s$ .  $i$  is upper bounded by the desired degree of correction required in the picture since the length of the remaining error streaks in the corrected received picture grows linearly with  $i$ .  $b$  is lower bounded by the accuracy of the correction. If  $b$  is less than the total number of bits in the transmitter estimate,  $b_T$ , (in Fig 5  $b = 4$  and  $b_T = 6$ ), then the correction algorithm requires that after every  $i$  delta modulated bits have been sent, the 'b' bit PCM estimate will be sent to the receiver. The next lowest significant bit present in the transmitters estimate and the receivers estimate will be momentarily set to one. All bits of lesser significance in both the transmitters and receivers estimate will be momentarily reset to zero. This has the effect of introducing an error into the estimates of the transmitter and receiver. It can be shown that the amplitude of the error is proportional to  $2^{-b-1}$  and the error will last for  $M$  delta modulator samples where  $M$  is upper bounded by

$$\sum_{n=0}^M 1.5^n < 2^{b_T - b - 1}$$

simplifying yields:

$$M < 2.5 \ln (2^{b_T - b - 2} + 1) \quad (6)$$

For the pictures shown in Fig 6  $b_T = 6$ ,  $b = 4$  and from Eq 6,  $M = 2$ . Thus, the error introduced by the error correction algorithm will correct itself in two delta modulator samples or one-half a pixel and have a maximum amplitude of two quantization levels. Such a small error of such short duration cannot be seen in the pictures of Fig 6.

## PART II.

The second part of the error correction algorithm detects the remaining error streaks left by the first part of the correction algorithm and replaces the streak by the average intensity of the adjacent lines. This is accomplished by comparing each pixel on a line with the pixel above and below. If the absolute value of the pixel in the middle exceeds both the pixel above and below by a prescribed threshold, the pixel will be replaced by the average value of the pixels above and below.

The setting of the threshold has an effect on the quality of the corrected picture. If the threshold is set too low, the picture will lose definition because too many pixels which are not part of an error streak will be replaced by the average value of the pixels above and below. If, on the otherhand, the threshold is set too high many error streaks will go undetected. Using a threshold setting of one-eighth the maximum value of a pixel, the picture in Fig 6c was transformed to that shown in Fig 6d. Note that some error streaks are still present in Fig 6d, and a slight loss of definition is observed. However, the general quality of the picture has been increased by the removal of the most severe error streaks.

### CONCLUSIONS

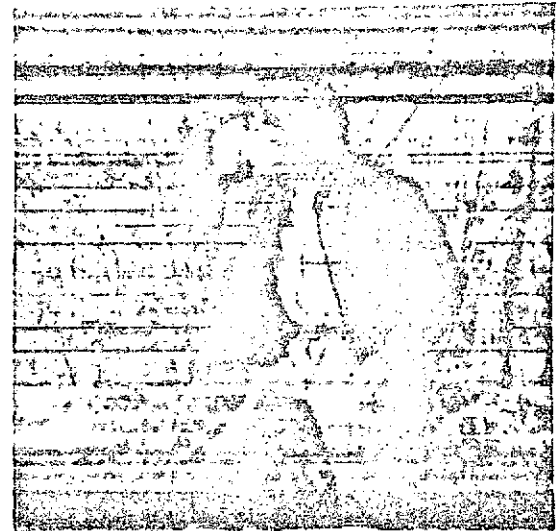
Delta modulators can be used to encode video signals. The pictures that result from these signals are similar in quality to pictures that result from PCM encoders. The problem of channel errors causing "picture streaks" in delta modulated encoded video signals has been minimized by the algorithm presented in this paper.

## REFERENCES

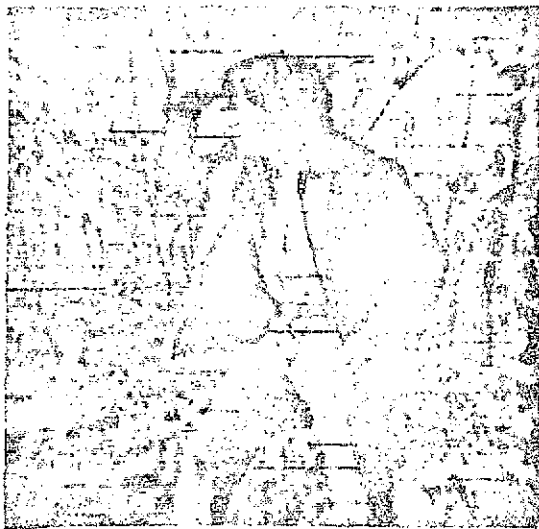
1. D.J. Connor, "Techniques for Reducing the Visibility of Transmission Errors in Digitally Encoded Video Signals," IEEE Transactions on Communications, Vol. COM-21, June 1973, pp. 695-706.
2. M.Z. Ali, I. Paz, N.R. Scheinberg, D.L. Schilling, "A Technique for Correcting Transmission Errors in Delta Modulation Channels," International Conference on Communications ICC 74, June 1974, Minneapolis Minnesota, pp.44F1- 44F6.
3. C.L. Song, J. Garodnick, D.L. Schilling, "A Variable Step Size Robust Delta- Modulator," IEEE Transactions on Communications, Vol. COM-19, December 1971, pp. 1033-1099.



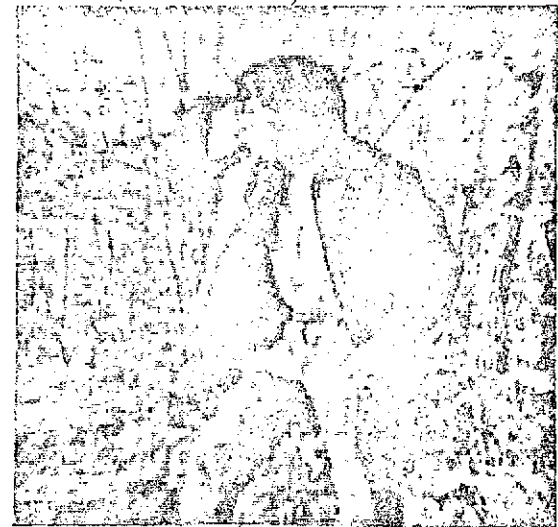
(a) DELTA MOD ENCODED VIDEO SIGNAL



(b) DELTA MOD ENCODED VIDEO  
SIGNAL WITH A CHANNEL  
ERROR RATE OF  $10^{-3}$  ERRORS/BIT



(c) PICTURE (b) AFTER FIRST  
PART OF THE CORRECTION  
ALGORITHM HAS BEEN APPLIED



(d) PICTURE (b) AFTER THE COMPLETE  
CORRECTION ALGORITHM HAS BEEN  
APPLIED

ORIGINAL PAGE IS  
OF POOR QUALITY

FIG 6 THE EFFECTS OF ERRORS AND ERROR CORRECTION ON DELTA MOD  
ENCODED PICTURES





PCM 5 BITS/PIXEL



PCM 4 BITS/PIXEL



PCM 3 BITS/PIXEL



DELTA MOD 5 BITS/PIXEL



DELTA MOD 4 BITS/PIXEL



DELTA MOD 3 BITS/PIXEL

ORIGINAL PAGE IS  
OF POOR QUALITY

FIG 2, 3 A COMPARISON OF PCM AND DELTA MOD ENCODED PICTURES

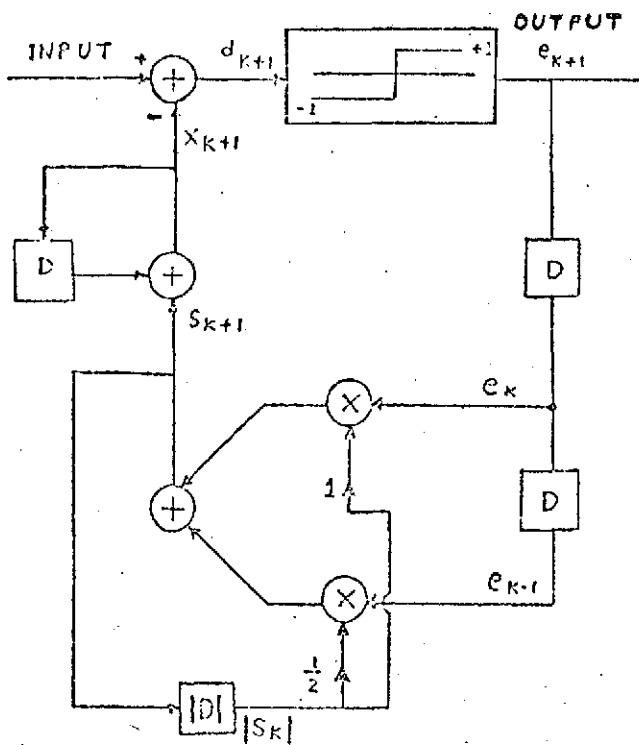


Fig. 1 Song Delta Modulator

ORIGINAL PAGE IS  
OF POOR QUALITY

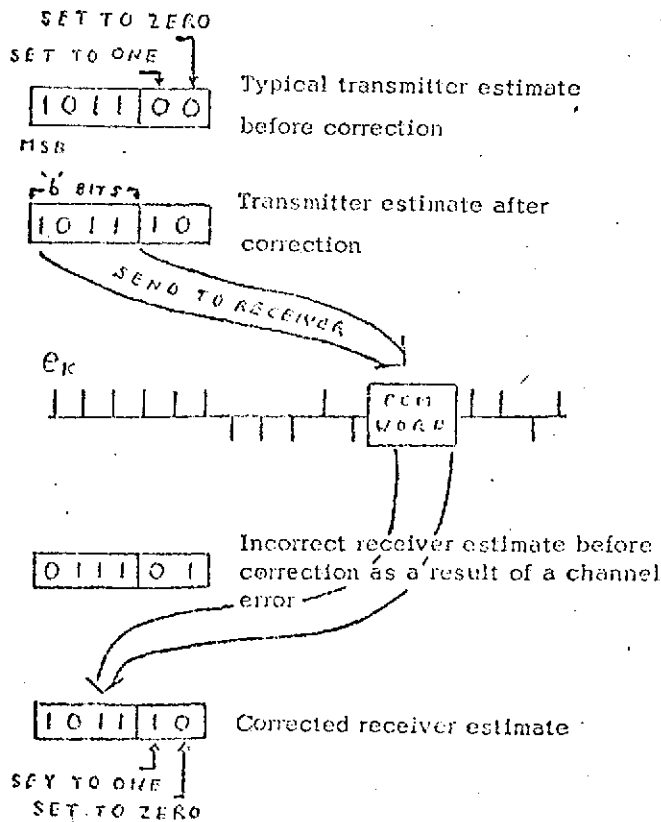


Fig. 5 Part 1 of the error correction algorithm

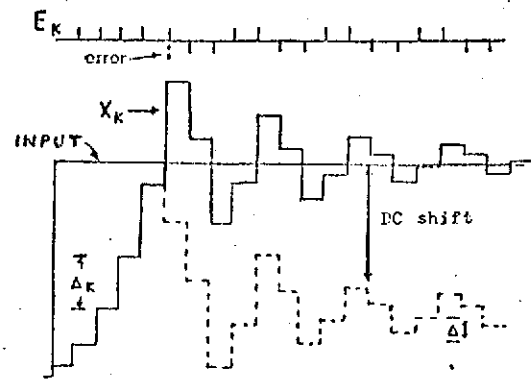


Fig. A No change in step size,  $\Delta_k$

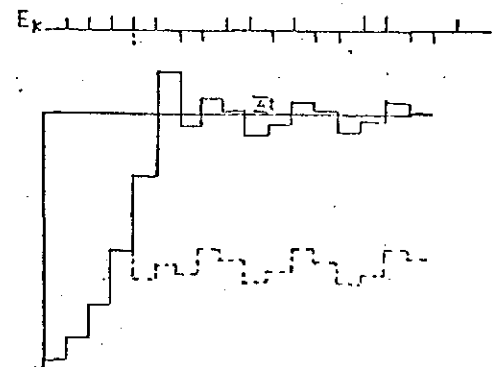


Fig. B Decrease in step size,  $\Delta_k$

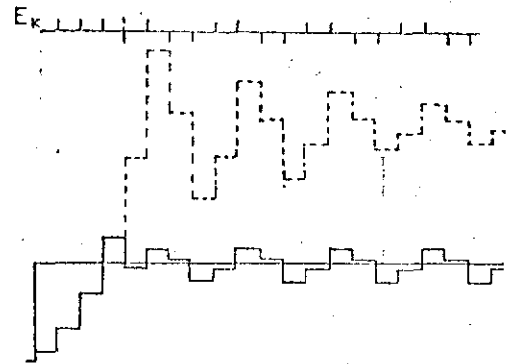


Fig. C Increase in step size,  $\Delta_k$

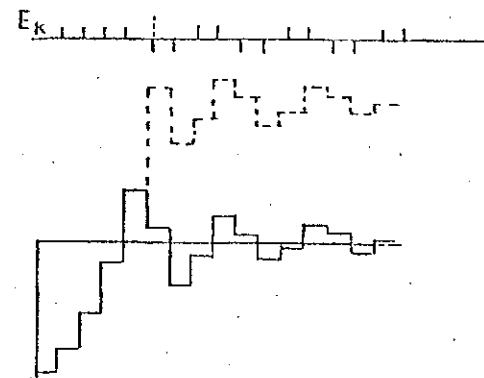


Fig. D No change in step size,  $\Delta_k$

Fig. 4 The effects of channel errors on the step size,  $\Delta_k$ , and the estimate,  $X_k$ , of a Song delta modulator